

1-1-2004

Ambiente GUI interactivo para el manejo de Procesamiento Digital de Señales (PDS)

José Edwin Torres Castro
Universidad de La Salle, Bogotá

Follow this and additional works at: https://ciencia.lasalle.edu.co/ing_electronica

Citación recomendada

Torres Castro, J. E. (2004). Ambiente GUI interactivo para el manejo de Procesamiento Digital de Señales (PDS). Retrieved from https://ciencia.lasalle.edu.co/ing_electronica/486

This Trabajo de grado - Pregrado is brought to you for free and open access by the Facultad de Ingeniería at Ciencia Unisalle. It has been accepted for inclusion in Ingeniería Eléctrica by an authorized administrator of Ciencia Unisalle. For more information, please contact ciencia@lasalle.edu.co.

**AMBIENTE GUI INTERACTIVO PARA EL MANEJO DE PROCESAMIENTO
DIGITAL DE SEÑALES (PDS)**

JOSÉ EDWIN TORRES CASTRO

**UNIVERSIDAD DE LA SALLE
FACULTAD DE INGENIERÍA ELÉCTRICA
BOGOTÁ D. C.**

2004



UNIVERSIDAD DE LA SALLE
FACULTAD DE INGENIERÍA ELÉCTRICA

**AMBIENTE GUI INTERACTIVO PARA EL MANEJO DE PROCESAMIENTO
DIGITAL DE SEÑALES (PDS)**

JOSÉ EDWIN TORRES CASTRO



UNIVERSIDAD DE LA SALLE
FACULTAD DE INGENIERÍA ELÉCTRICA
BOGOTÁ D. C.

2004

JOSÉ EDWIN TORRES CASTRO

**AMBIENTE GUI INTERACTIVO PARA EL MANEJO DE PROCESAMIENTO
DIGITAL DE SEÑALES (PDS)**

JOSÉ EDWIN TORRES CASTRO

**Monografía para acceder al título de
Ingeniero Electricista**

**Director
ING. Carlos Sanchez**



**UNIVERSIDAD DE LA SALLE
FACULTAD DE INGENIERÍA ELÉCTRICA
BOGOTÁ D. C.
2004**



Nota de aceptación

ING. CARLOS SÁNCHEZ

Director de proyecto

ING. DORIS HERNÁNDEZ D.

Jurado

ING. GUSTAVO ARCINIEGAS

Jurado

Bogotá D. C., _____
Día Mes Año

JOSÉ EDWIN TORRES CASTRO



Ofrezco este trabajo en forma especial a Dios, por llevarme de su mano siempre hasta donde he querido llegar, a mis padres, hermanos, esposa e hijo que confiaron en mi elección de ser Ingeniero Electricista, por su comprensión, apoyo y paciencia durante mi desarrollo personal y profesional.

José Edwin Torres Castro



AGRADECIMIENTOS

Expreso mis sentidos agradecimientos a:

LA UNIVERSIDAD DE LA SALLE por su gratificante acogida durante los años requeridos para conquistar el título profesional conjugado con el crecimiento espiritual haciéndome persona íntegra.

A los brillantes profesores por el acierto en despejar mi camino profesional como ingeniero electricista con sus conocimientos, laboratorios y herramientas técnicas para hacer realizable mi ejercicio profesional.

Carlos Sánchez, Ingeniero eléctrico, director de la tesis por su cordura en la orientación del desarrollo temático confianza y colaboración desinteresada en desarrollo de la presente tesis.

Luis H. Correa, Ingeniero Electricista, quien otorgándome alta dosis de confianza avalo el proyecto de tesis.

Javier Enrique Tavera, Ingeniero Mecatronico, por su colaboración y aporte incondicional a la investigación y realización del presente trabajo.

Saúl Díaz, Ingeniero Electrónico, por su valiosa colaboración y aporte en la investigación y realización del presente trabajo de tesis.



RESUMEN

UNIVERSIDAD DE LA SALLE FACULTAD DE INGENIERÍA ELÉCTRICA

AMBIENTE GUI INTERACTIVO PARA EL MANEJO DE PROCESAMIENTO DIGITAL DE SEÑALES (PDS)

El propósito de esta tesis es acompañar a los estudiantes de ingeniería eléctrica en el proceso académico y cognoscitivo específicamente tratado dentro del área del procesado digital de señales, considerando a MATLAB en una de las herramientas mas utilizadas y fuertes del mercado en la manipulación de datos matemáticos en diversas áreas, para nuestro propósito (PDS).

La creación de un ambiente interactivo dentro de una plataforma robusta pero a la vez complicada puede llegar a ser una herramienta sencilla para el aprendizaje de diversos temas. En el área de PDS es fundamental el análisis grafico del comportamiento de las señales, por esto se enfatiza en el desarrollo del ambiente GUI.

En este trabajo se recopila información básica sobre el procesado de señales y sobre MATLAB, generando finalmente un compendio de talleres prácticos para la corroboración del conocimiento adquirido.

La información al respecto de PDS y MATLAB es muy amplia y compleja, no obstante se seleccionaron algunos ítems sencillos y prácticos para la inicialización y contextualización en el área de PDS y en la plataforma matemática de MATLAB.



CONTENIDO

	Pág.
RESUMEN	
INTRODUCCIÓN	1
1. MATLAB	3
1.1. Presentación del programa	4
1.2. Uso del HELP	5
1.3. Fundamentos de las interfases graficas con MATLAB	6
2. CONCEPTOS BASICOS DE PROCESADO DIGITAL DE SEÑALES	7
2.1. Señales	7
2.2. Espectro de una señal	8
2.3. Tipos de señales	8
2.3.1. Señales continuas y discretas	8
3. Componentes de las señales	9
3.1 Energía	9
3.2. Potencia	9
3.3. Corriente continua o DC	9
3.4. Corriente alterna o AC	9
3.5. Valor eficaz o rms	10
4. Transformada de fourier continua	10
4.1. Series de fourier	10



4.2. Transformada de fourier	10
5. Transformada discreta de fourier	10
5.1. FFT (transformada rápida de fourier)	11
6. Transformada Z	11
7. Convolución	12
8. Filtros digitales	12
3. DESARROLLO MATEMATICO DEL PDS	13
1. Transformada de laplace	13
2. Transformada de fourier	15
3. Transformada Z	20
4. Convolución	22
5. Filtros digitales	26
4. DISEÑO DE GUI	28
4.1. Antecedentes	28
4.2. Principios de diseño	28
4.2.1. Simplicidad	29
4.2.1.1. Reducción del área de trabajo	29
4.2.2. Familiaridad	31
4.3. Proceso de diseño	31
4.3.1. Definición de funciones y trazado de la GUI	33
5. IMPLEMENTACION DE GUI	35



5.1. Antecedentes	35
5.2. Editor de propiedades	35
5.3. Panel de control	37
5.4. Editor de llamadas de función	41
5.5. Herramientas de alineación	42
5.6. Editor de menú	42
6. COMPONENTES DEL AMBIENTE GUI	44
6.1. Características	44
6.2. Detalles y explicación de las GUIs	45
6.2.1. Componentes de las señales	45
6.2.2. Convolución	45
6.2.3. Transformada de fourier continua	46
6.2.4. Transformada de fourier discreta	46
6.2.5. Transformada de laplace	47
6.2.6. Transformada Z	47
6.2.7. Filtros digitales	48
CONCLUSIONES	49
BIBLIOGRAFÍA	50
APÉNDICE A.	51
APÉNDICE B.	55
APÉNDICE C.	65
ANEXO A	67



FIGURAS

	Pág.
Figura 1.1 Ventanas de MATLAB	4
Figura 1.2 Menú Help de MATLAB	5
Figura 1.3 Ventana inicial de Help Windows	5
Figura 1.4 Ventana inicial de Help Desk	5
Figura 2.1 Señal tiempo continuo	8
Figura 2.2 Señal tiempo discreto	8
Figura 3.1 Grafica en MATLAB, ejercicio de transformada de laplace	15
Figura 3.2 Grafica en MATLAB, ejercicio de transformada de fourier (2.2)	18
Figura 3.3 Grafica en MATLAB, ejercicio de transformada de fourier (2.3)	19
Figura 3.4 Grafica en MATLAB, ejercicio de transformada Z	22
Figura 3.5 Grafica en MATLAB, ejercicio de convolución	26
Figura 3.6 Grafica en MATLAB, ejercicio de filtros digitales (5.2)	27
Figura 4.1 Simplicidad, consistencia y familiaridad	28
Figura 4.2 Ventanas separadas	30
Figura 4.3 Fusión en una sola ventana	30
Figura 4.4 Bosquejo inicial	33
Figura 4.5 Segundo Bosquejo	34
Figura 5.1 Editor de propiedades	36



Figura 5.2 Herramientas de GUIDE	38
Figura 5.3 Objetos disponibles desde el panel de control	39
Figura 5.4 Activación de figura	40
Figura 5.5 Editor de llamadas de función	41
Figura 5.6 Herramientas de alineación	42
Figura 5.7 Editor de menú	43
Figura 6.1 Ventana de presentación	44
Figura 6.2 GUI componentes de las señales	45
Figura 6.3 GUI convolución	45
Figura 6.4 GUI transformada de fuorier continua	46
Figura 6.5 GUI transformada de fourier discreta	46
Figura 6.6 GUI transformada de laplace	47
Figura 6.7 GUI transformada Z	47
Figura 6.8 GUI filtros digitales	48



ANEXOS

Anexo A. Disco donde se encuentra la carpeta de AMBIENTE GUI INTERACTIVO.



GLOSARIO

BREAK comando para terminar la ejecución de un lazo.

DOC este comando seguido del nombre de un comando muestra la información correspondiente a través de Netscape Navigator o Internet Explorer.

DTF transformada de fourier discreta.

END comando para cerrar oraciones precedidas por los comandos for, if y while.

ESPECTRO conjunto de frecuencias que constituyen la señal.

FOR comando utilizado para repetir un determinado numero de veces un grupo de oraciones del programa.

FFT transformada rápida de fourier.

FIR respuesta de impulso finito.

GET produce un listado de las propiedades y de sus valores.

GUI interfase grafica creada por un usuario.

HANDLES es un identificador único de cada objeto de Matlab.

HELP comando utilizado para buscar información en Matlab.

IF comando para ejecutar un grupo de oraciones de programa condicionalmente.

IIR respuesta de impulso infinito.

LENGTH comando que produce la longitud del vector.

MATLAB software interactivo que se usa en varias áreas de ingeniería y aplicaciones científicas.

PDS procesamiento digital de señales.

SET comando que lista en pantalla todas las propiedades de los objetos.

SIZE comando que nos indica el numero de columnas y lineas de la matriz.



WHILE comando para ejecutar un numero indefinido de veces el programa hasta que se cumpla cierta condición.

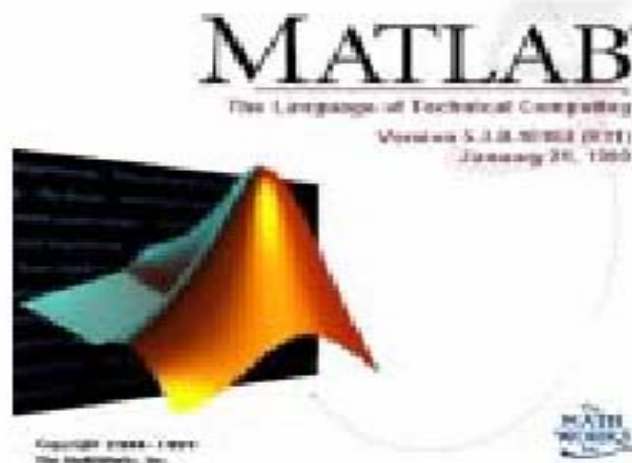
WHO comando que muestra el nombre de todas las variables creadas en Matlab.



INTRODUCCIÓN

MATLAB a lo largo de su existencia se ha ido popularizando y ganando la confianza de estudiantes, investigadores, técnicos e ingenieros; porque sus características incluyen facilidades de graficación inmediata, trabajo interactivo, funciones prediseñadas, posibilidad de añadir funciones escritas por usuarios y una programación simple. El paquete provee herramientas útiles para interactuar con programas externos y conjuntos de datos, así como la facilidad de llevar un registro de las variables utilizadas, que más tarde podrían emplearse como reporte técnico. Existen varias versiones de **MATLAB**, la más sencilla la de estudiante, que puede ser complementada con diversas cajas de herramientas diseñadas para campos avanzados y especializados en cierta aplicación, que se encuentran en la versión profesional del programa.

MATLAB es un poderoso lenguaje de programación de alto nivel para cálculos científicos. Es un programa sencillo de aprender y utilizar para la resolución de problemas numéricamente complejos. **MATLAB** consiste en funciones que son creadas directamente en el intérprete (ventana de comando) o accesibles como archivos-M (*M-files*), donde cada uno contiene una secuencia de comandos que ejecutan un cierto algoritmo. Un algoritmo nuevo, puede escribirse como un programa que contenga algunas de estas funciones preestablecidas y puede ser guardado como otro archivo-M; que para operar, las funciones en que se apoya deben ser accesibles a través de la trayectoria de archivos en **MATLAB**.





La creación de una Interfase Gráfica al Usuario basada en **MATLAB** (GUI), no importando qué tan grande o pequeña, simple o extravagante sea; es **única entidad**, con **cierta personalidad** para ayudar a un **usuario específico** a realizar una **tarea específica**. El enfoque de esta tesis es moldear todos éstos aspectos, sabiendo de antemano que el usuario específico es cualquier persona interesada en el estudio del procesamiento digital de señales, que cuente con una noción previa de la materia para reafirmar sus conocimientos y desarrollar capacidades por medio de la repetida práctica sobre el **AMBIENTE GUI INTERACTIVO** creado en **MATLAB**, el usuario específico puede o no haber previamente tenido contacto con el programa **MATLAB**, pues más que llegar a desarrollar algún código en ése lenguaje trabajará directamente con un ambiente diseñado de tal manera que no se le dificulte su utilización debido a que está compuesto por elementos obvios al manejo y sencillos de maniobrar. La tarea específica a realizar depende del programa que se seleccione dentro de la gama que ofrece el **ambiente GUI**.

La concepción de esta tesis se fundó a partir de la idea: Para qué utilizar varias ventanas si con una es suficiente para agrupar ordenadamente todos los objetos necesarios? Esta oración en este punto podría no tener mucho significado, pero adentrándose en los fines perseguidos en el proyecto y analizándola desde otra perspectiva, es una frase llena de sentido.

Debido a que las buenas herramientas están hechas para un trabajo placentero, la finalidad fue crear un conjunto de herramientas que por medio de la investigación satisficieran las necesidades de un usuario final otorgándole todos los elementos para que se adentrara en el mundo del Procesado Digital de Señales de una manera agradable, sencilla, interactiva, amigable y elegante.

Con el fin de que las interfases graficas de usuario creadas en Matlab satisfagan las necesidades del usuario, se realizaron una serie de practicas de los conceptos básicos de de procesamiento digital de señales la cual nos ayudaran a introducirnos a este tema de una manera mas agradable y eficaz.

Agradable, porque es algo que se sale de lo común y motiva al usuario induciéndolo a explorar los programas con inquietud, evitándole a trabajar siguiendo unos esquemas rígidos que a la postre resultan frios y poco gratificantes.

Eficaz, porque le da la libertad al operador de trasladarse o pasarse seguro de un programa a otro, sin un orden rígido y las veces que desee. Analizando resultados generados a partir de las condiciones de entrada establecidas por él.



1. MATLAB

¿Qué es MATLAB? Es un software interactivo que ha sido usado recientemente en varias áreas de ingeniería y aplicaciones científicas. No es un lenguaje computacional en el sentido “normal”, pero trabaja como tal. Un aspecto atractivo de MATLAB es que es relativamente fácil de aprender. El poder de MATLAB es representado por la longitud y simplicidad de su código. Por ejemplo, una página de código de MATLAB puede ser equivalente a muchas páginas de códigos fuentes de otros lenguajes computacionales. Para los cálculos numéricos, MATLAB usa colecciones de subrutinas matemáticas científicamente bien escritas como LINPACK y EISPACK.

MATLAB es el nombre abreviado de “MATrix LABoratory”. MATLAB es un programa para realizar cálculos numéricos con **vectores** y **matrices**. Como caso particular puede también trabajar con números escalares, tanto reales como complejos. Una de las capacidades más atractivas es la de realizar una amplia variedad de **gráficos** en dos y tres dimensiones. MATLAB tiene también un lenguaje de programación propio. Esta tesis hace referencia a la versión 5.3 de este programa, aparecida a comienzos de 1999.

En general, MATLAB es una poderosa herramienta para manejos de vectores y matrices. Estos arreglos no requieren dimensionamiento, por lo que se resuelven muchos problemas de computación técnica, sobre todo la formulación de vectores y matrices se hace en un tiempo corto, que no se podría hacer con lenguajes como Fortran o C.

MATLAB durante más o menos 15 años de vida (1984), tiene millones de usuarios. En las universidades se usa como herramienta para cursos introductoria y avanzada de matemáticas, ingeniería y ciencias. En la industria, es una herramienta para la investigación de alta productividad, desarrollo y análisis.

MATLAB se puede arrancar como cualquier otra aplicación de **Windows 95/98/NT**, clicando dos veces en el icono correspondiente o por medio del menú **Start**). Al arrancar MATLAB se abre una ventana del tipo de la indicada en la Figura 1.

En la ventana inicial se sugieren ya algunos comandos para el usuario inexperto que quiere echar un vistazo a la aplicación. En dicha ventana aparece también el **prompt** característico de MATLAB (»). Esto quiere decir que el programa está preparado para recibir instrucciones.



1.1. Presentación del programa

MATLAB opera en tres diferentes ventanas desplegadas en la pantalla de la computadora. Las cuales son *Ventana de comandos* (Command Windows), *Ventana de figura* (Figure Windows), y ventana del Editor (Editor Windows). La ventana de Comando se titula en el encabezado **Command**, la ventana de Figura se titula **Figure No. 1**, y la ventana de Editor muestra el nombre del actual archivo-M que se encuentra abierto, o en su caso **Untitled** si es un archivo nuevo que se comienza a trabajar o se encuentra en edición. La ventana de Comando muestra el apuntador **>>** indicando que el programa esta listo para ejecutar comandos de MATLAB. Si existieran resultados arrojados por la resolución de comandos, se muestran en la ventana de Comando. Escribir un punto y coma al final de cada oración, asegura que el resultado generado por el comando, no se imprima en la ventana. Si posteriormente se necesitara saber el valor de la variable, solo basta teclear su nombre sin *punto y coma* en dicha ventana para que se despliegue su valor. También desde la ventana de comando es posible correr los programas y funciones previamente creados y guardados como archivos-M. Todos los gráficos generados por los comandos de trazado, aparecen en la ventana de Figura. Cualquier tipo de archivo-M ya sea nuevo o previamente creado, se puede echar a andar escribiendo el nombre del archivo en la ventana de Comando y dándole *ENTER*.

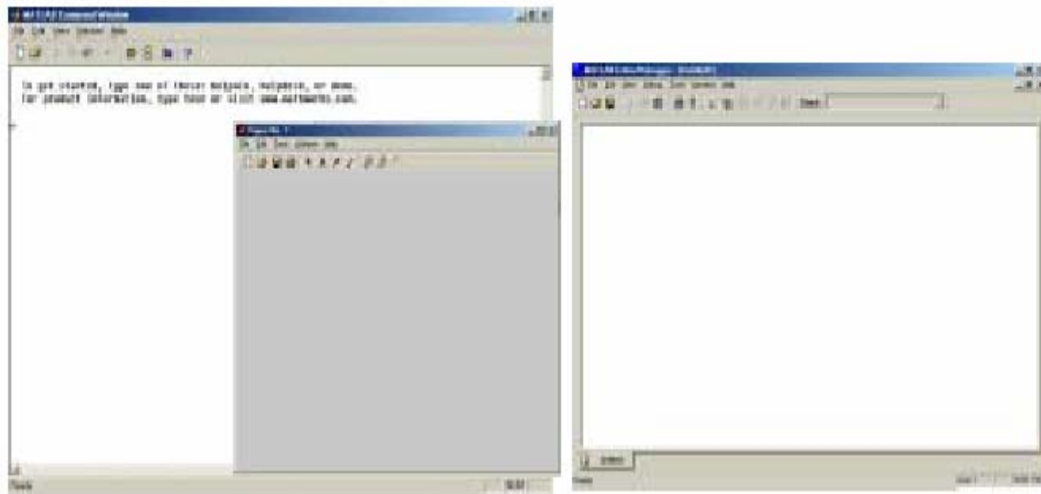


Figura 1.1 Ventanas de MATLAB.



1.2. Uso del HELP

MATLAB 5.3 dispone de un excelente **Help** con el que se puede encontrar la información que se desee. La Figura 1.2 muestra las distintas opciones que aparecen en el menú **Help**.

1. **Help Window**, Se abre la ventana de la Figura 1.3, en la que se puede buscar ayuda sobre la función o el concepto que se desee.

2. **Help Tips**. Ofrece ideas prácticas para utilizar la ayuda.

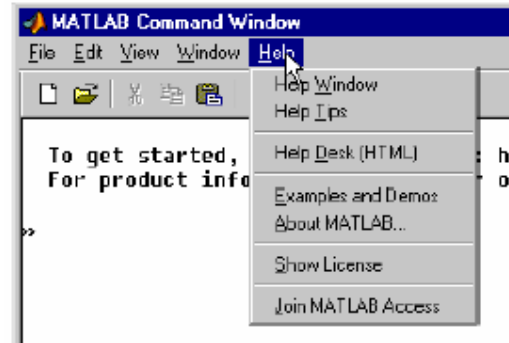


Fig. 1.2 Menu Help de Matlab

3. **Help Desk**. Se abre un browser de Internet (**Netscape Communicator**, en el caso de la Figura 1.4) que permite acceder a toda la información sobre MATLAB en formato HTML. Esta información es equivalente a los manuales impresos del programa. Desde la parte inferior de esta página, mediante el enlace **Online Manuals (in PDF)** se puede acceder a la versión ***.pdf** (*Portable Document Format*) de los manuales de MATLAB. Este formato es menos adecuado para consultar que el HTML, pero mucho más adecuado para imprimir y revisar luego sobre papel. El formato ***.pdf** requiere del programa gratuito **Adobe Acrobat Reader 5.0** o una versión superior.

4. **Examples and Demos**. Se abre una ventana que da acceso a un buen número de ejemplos resueltos con MATLAB, cuyos resultados se presentan gráficamente de diversas formas. Es bastante interesante recorrer estos ejemplos para hacerse idea de las posibilidades del programa. Es asimismo muy instructivo.

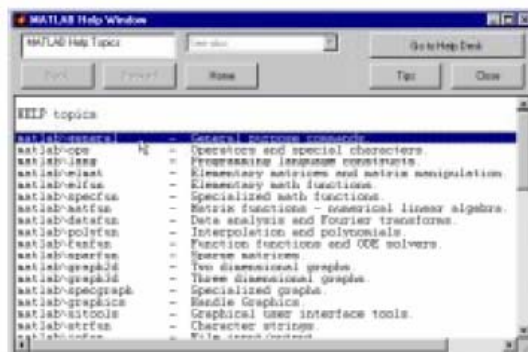


Fig. 1.3 Ventana inicial de **Help Window**.



Fig. 1.4 Ventana inicial de **Help Desk**.



Además, se puede también recurrir al **Help** desde la línea de comandos. Se aconseja hacer prácticas al respecto. Por ejemplo, obsérvese la respuesta a los siguientes usos del comando **help**:

- » **help**
- » **help lang**

El comando **helpwin** seguido de un nombre de comando muestra la información correspondiente a ese comando en la ventana **Help Window** (ver Figura 1.3), incluyendo también comandos similares sobre los que se ofrece ayuda.

El comando **doc** seguido de un nombre de comando muestra la información correspondiente a ese comando a través de **Netscape Navigator** o **Internet Explorer**, en formato HTML.

1.3. Fundamentos de las Interfaces Gráficas con MATLAB

MATLAB permite desarrollar de manera simple un conjunto de pantallas con botones, menús, ventanas, etc., que permiten utilizar de manera muy simple programas realizados en el entorno **Windows**. Este conjunto de herramientas se denomina **interface de usuario**. Las posibilidades que ofrece MATLAB 5.3 han mejorado mucho respecto a versiones anteriores, aunque no son muy amplias en comparación con otras aplicaciones de **Windows** como **Visual Basic**.

Para poder hacer programas que utilicen las capacidades gráficas avanzadas de MATLAB hay que conocer algunos conceptos básicos del funcionamiento de MATLAB. Aunque MATLAB dispone ahora de la herramienta GUIDE, que permite generar interfaces de usuario de una forma muy cómoda y sencilla, es conveniente conocer los fundamentos de lo que se está haciendo, e incluso ser capaz de hacerlo programando si ayudas. Por esta razón es importante que los lectores o estudiantes interesados en este tema se dirijan a los diversos manuales que tiene MATLAB para así aprender toda la estructura y funcionamiento de esta valiosa herramienta muy útil para estudiantes de ingeniería y personas que estén dedicadas a la investigación.



2. CONCEPTOS BÁSICOS DE PROCESADO DIGITAL DE SEÑALES (PDS)

1. Introducción

El procesamiento digital de señales comenzó a causa del deseo de los diseñadores de sistemas analógicos de simular el comportamiento de éstos antes de llevar a cabo la construcción de costosos prototipos. La herramienta para realizar la simulación era, por supuesto, el ordenador y fue de esta manera como se inició el desarrollo del procesamiento digital de señales. Es gracioso pensar que aquellos pioneros que trabajaron entre la década de los 50 y los primeros años de la de los 60 no tenían mucha idea de que su trabajo se expandiría hacia un área mucho mayor en la tecnología de la electrónica digital durante los años 80 y posteriores.

El procesamiento digital de las señales dependía de los procesadores digitales y de la mayor parte de las matemáticas o algoritmos que habían sido desarrollados desde los años 50. Una vez que los algoritmos estaban establecidos, los diseñadores comenzaron a buscar la arquitectura de procesador más eficaz para implementarlos. La primera meta fue conseguir simulaciones que se ejecutasen en un tiempo aceptable. No está claro cuando surgió la idea de utilizar computadores digitales para procesar señales reales, pero una vez hecho, los objetivos cambiaron súbitamente. La meta final es el procesamiento digital de las señales en tiempo real, esto es, realizar los cálculos en el instante mismo en que la señal llega al sistema. Los antiguos sistemas tan sólo eran capaces de almacenar la onda en memoria y procesarla más tarde.

2. Señales y Sistema

2.1. Señales

Las señales pueden describir una variedad muy amplia de fenómenos físicos, y aunque se pueden representar de muchas formas, en todo caso la información dentro de una señal está contenida en un patrón de variaciones de alguna forma.

Es decir, una señal no va a ser más que una función de una o unas variables independientes que contiene información acerca de la naturaleza o comportamiento de algún fenómeno. Así, por ejemplo, la señal de voz se representa de forma matemática por la presión acústica como una función del tiempo.



Hay dos tipos básicos de señales, de tiempo continuo y de tiempo discreto.

En el caso de las señales de tiempo continuo la variable independiente es continua y entonces estas señales están definidas para una sucesión continua de valores de la variable independiente.

De otra parte, las señales de tiempo discreto están sólo definidas en tiempos discretos y en consecuencia para estas señales, la variable independiente toma solo un conjunto de valores discretos.

2.2. Espectro de una señal

El espectro de una señal es el conjunto de frecuencias que constituyen la señal.

El ancho de banda es la anchura del espectro. Muchas señales tienen un ancho de banda infinito, pero la mayoría de la energía está concentrada en un ancho de banda pequeño.

2.3. Tipos de señales

2.3.1. Señales continuas -señales discretas

- Las **señales de tiempo continuo** son aquellas cuya variable independiente es continua y, por tanto, está definida para un conjunto continuo de valores de dicha variable. Dicho de otro modo, son aquellas cuya variable independiente pertenece al conjunto de los números reales.
- Las **señales de tiempo discreto** poseen solo definición para una sucesión discreta de valores; esto es, su variable independiente pertenece al conjunto de los números enteros.

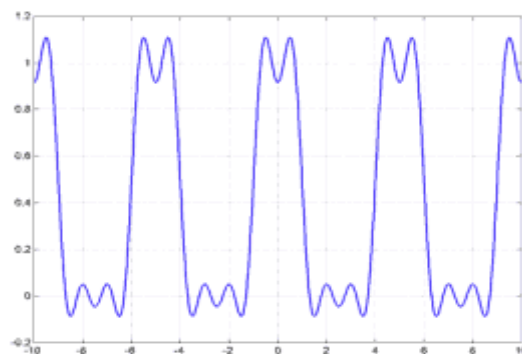


Fig. 2.1 Señal tiempo continuo

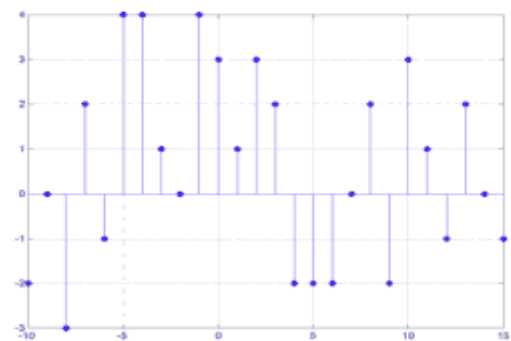


Fig. 2.2 Señal tiempo discreto



- Una señal discreta se obtiene a partir de una señal continua al hacer una secuencia de muestreo de esta señal, es importante que el muestreo sea periódico. Para mayor información sobre este tema ver teorema de muestreo.
- Para que una señal muestreada pueda ser procesada como una serie de números, es necesario asignarle un valor numérico específico, reduciendo con esto la resolución del sistema y aumentando el error. Este error se llama error de cuantización.

3. Componentes de las señales

3.1. Energía

Capacidad de un sistema para realizar un trabajo, con las propiedades de la conservación y la interconvertibilidad la cual se puede convertir en un trabajo útil cuando este experimenta una transformación a presión y temperatura constante.

3.2. Potencia

Es la energía que absorbe un dispositivo en la unidad de tiempo, la potencia es una magnitud que relaciona el trabajo realizado por un sistema y el tiempo empleado en realizarlo.

Su ecuación principal es $(P = V * I)$ la que se expresa en Watt

3.3. Corriente continua o D.C

La corriente continua (c.c.) es el flujo continuo de electricidad a través de un conductor entre dos puntos de distinto potencial. A diferencia de la corriente alterna (c.a.), en este caso, las cargas eléctricas circulan siempre en la misma dirección del punto de mayor potencial al de menor potencial. Aunque comúnmente se identifica la corriente continua con la corriente constante (por ejemplo la suministrada por una batería), es continua toda corriente que mantenga siempre la misma polaridad.

3.4. Corriente alterna o A.C

Se denomina Corriente alterna (CA ó AC en inglés) a la corriente eléctrica que cambia repetidamente de polaridad. esto es, su voltaje instantáneo va cambiando en el tiempo desde 0 a un máximo positivo, vuelve a cero y continúa hasta otro máximo negativo y así sucesivamente. La corriente alterna más comúnmente



utilizada, cambia sus valores instantáneos de acuerdo con la función trigonométrica seno, de ahí se denominación de corriente alterna senoidal.

3.5. Valor eficaz o rms

Se llama valor eficaz de una corriente alterna, al valor que tendría una corriente continua que produjera la misma potencia que dicha corriente alterna, al aplicarla sobre una misma resistencia.

Es decir, se conoce el valor máximo de una corriente alterna (I_0). Se aplica ésta sobre una cierta resistencia y se mide la potencia producida sobre ella.

4. Transformada de Fourier Continua

4.1. Series de Fourier

- ☀ Las series de Fourier describen señales periódicas como una combinación de señales armónicas (sinusoides).
- ☀ Con esta herramienta podemos analizar una señal periódica en términos de su contenido frecuencial o espectro.
- ☀ La potencia contenida en una señal puede evaluarse a partir de los coeficientes de su correspondiente serie de Fourier.

4.2. Transformada de Fourier

- 🕒 La transformada de fourier es la generalización de las series de fourier.
- 🕒 Con esta herramienta podemos analizar una señal no periódica como una señal de periodo infinito.

5. Transformada Discreta de Fourier

- ➡ EL DFT es una aproximación al espectro de la señal analógica original. Su magnitud se ve influenciada por el intervalo de muestreo, mientras que su fase depende de los instantes de muestreo.



5.1. FFT (Fast Fourier Transform)

El FFT es un algoritmo que consiste en simplificar enormemente el calculo del DTF introduciendo “**atajos**” matemáticos para reducir drásticamente el numero de operaciones para llegar a la transformada de Fourier.

Existen dos métodos importantes para desarrollar el FFT, son por medio de la decimación en el tiempo y la decimación en frecuencia.

6. Transformada Z

La transformada z de una señal de tiempo discreto $x[n]$ se define como:

$$x(z) = \sum_{n=-\infty}^{\infty} x[n] * z^{-n}$$

donde z es una variable compleja.

La transformada z de una señal $x[n]$ se denota por

$$x(z) = z\{x[n]\}$$

Mientras que la relación entre $x[n]$ y $X(z)$ se indica mediante

$$x[n] \xleftrightarrow{z} x(z)$$

Desde un punto de vista matemático, la transformada z es simplemente una representación alternativa de la señal. De este modo el coeficiente de z^{-n} , para una transformada determinada, es el valor de la señal en el instante n . Y por tanto, el exponente de z contiene la información necesaria para identificar las muestras de la señal.



7. Convolución

Es la operación matemática que permite obtener la salida Y de un sistema a partir de la entrada X y su función de transferencia H .

La convolución es una operación laboriosa en el dominio temporal, pero muy sencilla en el dominio frecuencial (un simple producto de variables complejas). Por esta razón, el procesamiento de señal digital acostumbra a realizarse en el dominio frecuencial.

8. Filtros digitales

El término *filtro digital* lo entenderemos como cualquier procesamiento realizado en una señal de entrada digital.

Un filtro digital es la implementación en hardware o software de una ecuación diferencia.

Ventajas de los filtros digitales

- Alta inmunidad al ruido
- Alta precisión (limitada por los errores de redondeo en la aritmética empleada)
- Fácil modificación de las características del filtro
- Muy bajo coste (y bajando)

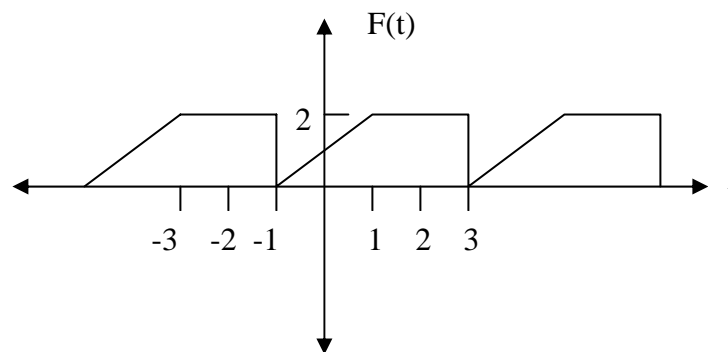
Por estas razones, los filtros digitales están reemplazando rápidamente a los filtros analógicos.



3. DESARROLLO MATEMÁTICO DEL PROCESADO DIGITAL DE SEÑALES (PDS)

1. TRANSFORMADA DE LAPLACE

Hallar la transformada de Laplace de la función periódica mostrada en la siguiente figura:



En el intervalo $-1 \leq t < 3$ la función se puede definir como:

$$f(t) = \begin{cases} \frac{t}{2} + \frac{1}{2}; & -1 \leq t < 1 \\ 2; & 1 \leq t < 3 \end{cases}$$

fuera del intervalo, se sabe que $f(t) = f(t+4)$, sabiendo que $T = 4$.

Se hace uso de: $\ell\{f(t)\} = \frac{1}{1 - e^{-st}} \int_0^T e^{-st} f(t) dt$

Transformada de una función periódica.

$$\ell\{f(t)\} = \frac{1}{1 - e^{-4s}} \int_0^4 e^{-st} f(t) dt$$



$$\begin{aligned} &= \frac{1}{1-e^{-4s}} \left[\int_{-1}^1 e^{-st} \left(\frac{t}{2} + \frac{1}{2} \right) dt + \int_1^3 e^{-st} 2dt \right] \\ &= \frac{1}{1-e^{-4s}} \left[\int_{-1}^1 e^{-st} \frac{t}{2} dt + \int_{-1}^1 e^{-st} dt + 2 \int_1^3 e^{-st} dt \right] \\ &= \frac{1}{1-e^{-4s}} \left[\frac{e^s}{s^2} - \frac{e^s}{s} - \frac{e^{-s}}{s^2} - \frac{e^{-s}}{s} + \frac{e^{-s}}{-2s} + \frac{e^s}{2s} - \frac{2e^{-3s}}{s} + \frac{2e^{-s}}{s} \right] \\ &= \frac{1}{1-e^{-4s}} \left[\frac{e^s}{s^2} - \frac{e^s}{2s} - \frac{e^{-s}}{s^2} - \frac{2e^{-3s}}{s} + \frac{1}{2} * \frac{e^{-s}}{s} \right] \\ &= \frac{1}{1-e^{-4s}} \left[\frac{2e^s - se^s - 2e^{-s} - 4se^{-3s} + se^{-s}}{2s^2} \right] \\ &= \frac{1}{1-e^{-4s}} \left[\frac{2e^s - 2e^{-s} + s(e^{-s} - e^s - 4e^{-3s})}{2s^2} \right] \end{aligned}$$

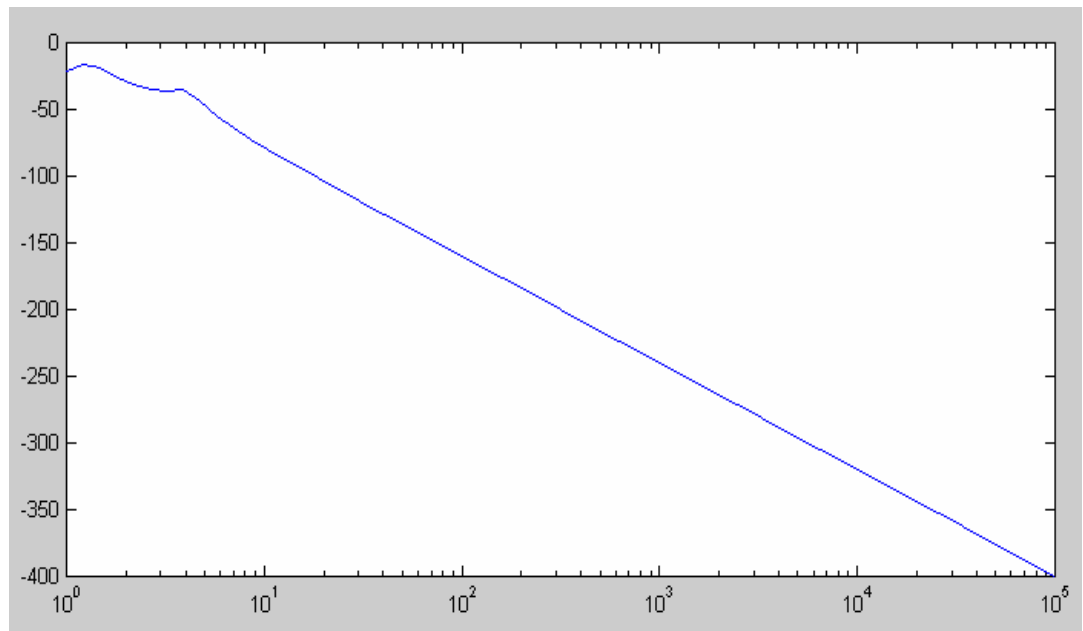


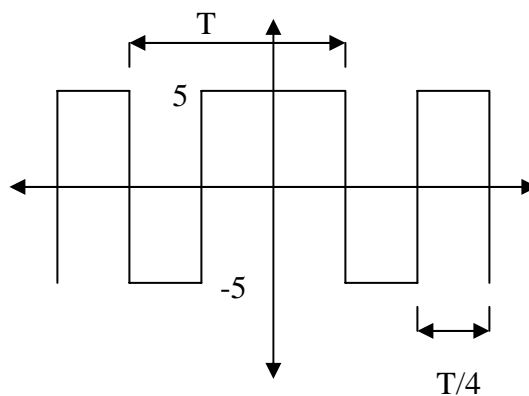
Fig. 3.1 grafica en Matlab, ejercicio de transformada de laplace

2. TRANSFORMADA DE FOURIER

2.1. Hallar la transformada de fourier para la siguiente señal en tiempo continuo:

Frecuencia: 400 Hz

2 periodos





$$C_k = \frac{1}{T_0} \int_{t_0}^{t_0+T_0} x(t) e^{-j2\pi K F_0 t} dt$$

$$C_k = \frac{1}{T_0} \left[\int_{-\frac{5T}{8}}^{\frac{3T}{8}} -5e^{-j2\pi K F_0 t} dt + \int_{\frac{3T}{8}}^{\frac{3T}{8}} 5e^{-j2\pi K F_0 t} dt \right]$$

$$C_k = \frac{-5}{T_0} \left[\frac{e^{-j2\pi K F_0 t}}{-j2\pi K F_0} \right]_{-\frac{5T}{8}}^{\frac{3T}{8}} + \frac{5}{T_0} \left[\frac{e^{-j2\pi K F_0 t}}{-j2\pi K F_0} \right]_{\frac{3T}{8}}^{\frac{3T}{8}}$$

$$C_k = \frac{5}{j2\pi K} \left[\cos\left(\frac{2\pi K(-3)}{8}\right) - isen\left(\frac{2\pi K(-3)}{8}\right) - \cos\left(\frac{2\pi K(-5)}{8}\right) + isen\left(\frac{2\pi K(-5)}{8}\right) - \right. \\ \left. \cos\left(\frac{2\pi K 3}{8}\right) + isen\left(\frac{2\pi K 3}{8}\right) + \cos\left(\frac{2\pi K(-3)}{8}\right) - isen\left(\frac{2\pi K(-3)}{8}\right) \right]$$

$$C_k = \frac{5}{j2\pi K} \left[\cos\left(\frac{-3}{4}\pi K\right) - isen\left(\frac{-3}{4}\pi K\right) - \cos\left(\frac{-5}{4}\pi K\right) + isen\left(\frac{-5}{4}\pi K\right) - \cos\left(\frac{3}{4}\pi K\right) \right. \\ \left. + isen\left(\frac{3}{4}\pi K\right) + \cos\left(\frac{-3}{4}\pi K\right) - isen\left(\frac{-3}{4}\pi K\right) \right]$$

$$C_k = \frac{5}{j2\pi K} \left[-2isen\left(\frac{-3}{4}\pi K\right) - \cos\left(\frac{5}{4}\pi K\right) + isen\left(\frac{-5}{4}\pi K\right) + isen\left(\frac{3}{4}\pi K\right) + \cos\left(\frac{3}{4}\pi K\right) \right]$$



$$C_k = \frac{5}{j2\pi K} \left[3 \operatorname{isen} \left(\frac{3}{4} \pi K \right) - \cos \left(\frac{5}{4} \pi K \right) + \operatorname{isen} \left(\frac{-5}{4} \pi K \right) + \cos \left(\frac{3}{4} \pi K \right) \right]$$

$$C(0) = \frac{5}{T} \int_{\frac{-5T}{8}}^{\frac{-3T}{8}} dt + \frac{5}{T} \int_{\frac{-3T}{8}}^{\frac{3T}{8}} dt$$

$$\frac{5}{T} \left[\frac{-3T}{8} + \frac{5T}{8} \right] + \frac{5}{T} \left[\frac{3T}{8} + \frac{3T}{8} \right]$$

$$\frac{5}{T} \left(\frac{2T}{8} \right) + \frac{5}{T} \left(\frac{6T}{8} \right) = \frac{10T}{8T} + \frac{30T}{8T} = \frac{40}{8} = 5$$

$$C(0) = 5$$

2.2. Hallar la transformada de fourier para la siguiente secuencia (FFT) para señal discreta.

$$x(n) = \begin{cases} A; 0 \leq n \leq L-1 \\ 0; \text{otro} \rightarrow \text{caso} \end{cases}$$

$$\sum_{n=-\infty}^{\infty} |x(n)| = \sum_{n=0}^{L-1} |A| = L|A| < \infty$$

Es decir que la energía de la señal es finita y $x(n)$ es sumable así que su transformada existe.



$$x(w) = \sum_{n=0}^{L-1} A e^{-jwn} = A \frac{1 - e^{-jwL}}{1 - e^{-jw}} = A e^{-j(w/2)(L/2)} \frac{\text{sen}(wL/2)}{\text{sen}(w/2)}$$

para $w = 0$ $|x(0)| = AL$

y $w \neq 0$ $|x(w)| = |A| \left| \frac{\text{sen}(wL/2)}{\text{sen}(w/2)} \right|$

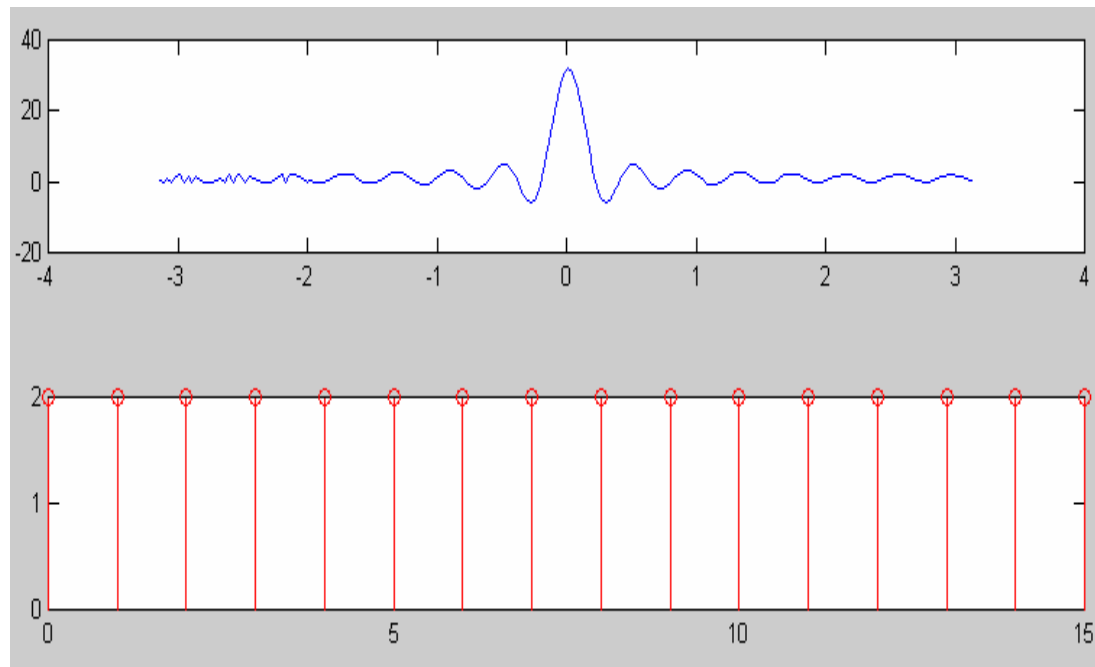
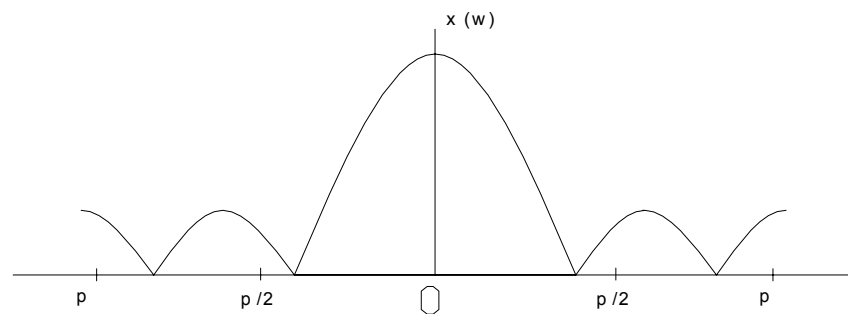


Fig. 3.2 grafica en Matlab, ejercicio de transformada de fourier (2.2)



2.3. Hallar la transformada de fourier discreta (DFT) para la secuencia del ejercicio anterior (2.2).

Determinar la DFT de N puntos para $N \geq L$

$$x(w) = \sum_{n=0}^{L-1} x(n) e^{-jwn} = \dots e^{-jw(L-1)/2} \frac{\text{sen}(wL/2)}{\text{sen}(w/2)}$$

La transformada de fourier discreta es simplemente $x(w)$ pero calculando no para w continuo sino en N frecuencias $w_k = 2\pi k/N$ con $k = 0, 1, \dots, N-1$ entonces:

$$x(k) = e^{-j\pi k(L-1)/N} \frac{\text{sen}(\pi kL/N)}{\text{sen}(\pi k/N)}$$

Si $N = L$ la respuesta grafica seria un impulso en frecuencia y aunque la información importante para que el calculo sea mas real y la señal se pueda recuperar es indispensable que el muestreo en frecuencia N sea mayor al numero de muestras en el tiempo L, es decir $N > L$.

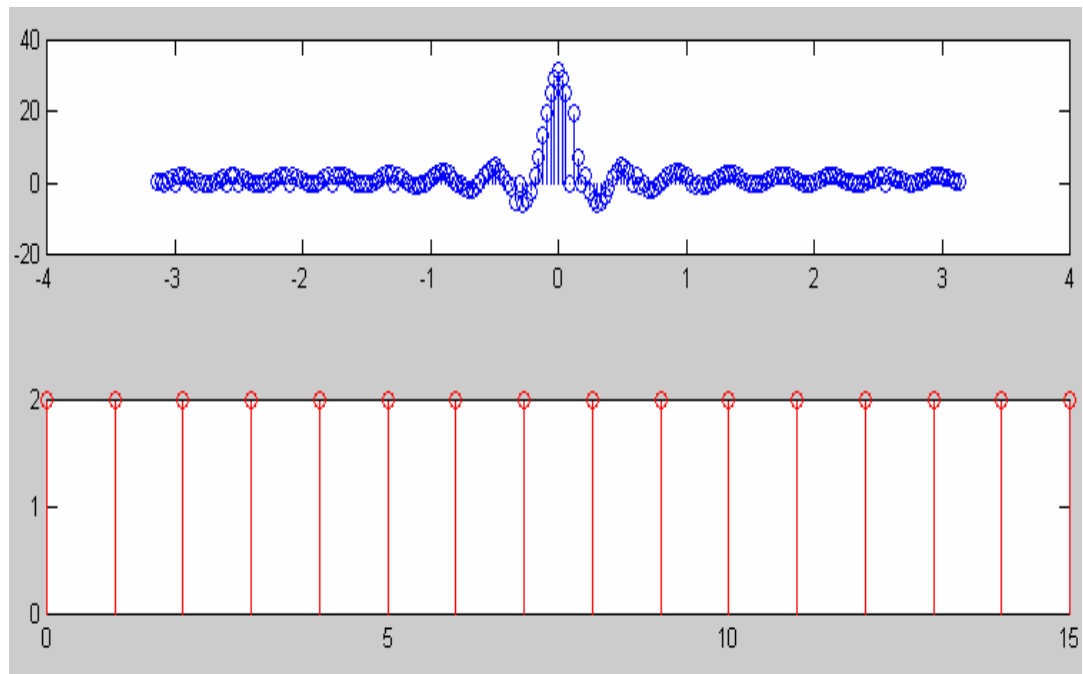


Fig. 3.3 grafica en Matlab, ejercicio de transformada de fourier (2.3)



3. TRANSFORMADA Z

Hallar la transformada Z de $a^n \text{sen}(W_0 n)$

- a. primero hallar la transformada Z de $\text{sen}(W_0 n)$, para eso utilizo la formula de euler.

$$\begin{aligned} e^{-jW_0 n} &= \cos W_0 n - j \text{sen} W_0 n \\ e^{jW_0 n} &= \cos W_0 n + j \text{sen} W_0 n \\ \hline e^{-jW_0 n} - e^{jW_0 n} &= 0 - 2j \text{sen} W_0 n \\ \text{sen} W_0 n &= \frac{e^{jW_0 n} - e^{-jW_0 n}}{2j} \end{aligned}$$

$$\begin{aligned} Z\{\text{sen} W_0 n\} &= z \left\{ \frac{e^{jW_0 n} - e^{-jW_0 n}}{2j} \right\} = z \left\{ \overbrace{\frac{e^{jW_0 n}}{2j}}^{\text{Propiedad de la linealidad}} - \overbrace{\frac{e^{-jW_0 n}}{2j}}^{\text{inealidad}} \right\} \\ &= \frac{1}{2j} z \{e^{jW_0 n}\} - \frac{1}{2j} z \{e^{-jW_0 n}\} \\ &= \frac{1}{2j} \left\{ \frac{1}{1 - e^{jW_0 n} z^{-1}} \right\} - \frac{1}{2j} \left\{ \frac{1}{1 - e^{-jW_0 n} z^{-1}} \right\} \\ &= \frac{1}{2j} \left\{ \frac{1}{1 - e^{jW_0 n} z^{-1}} - \frac{1}{1 - e^{-jW_0 n} z^{-1}} \right\} \end{aligned}$$



$$= \frac{1}{2j} \left\{ \frac{-e^{-jW_0 n} z^{-1} + e^{jW_0 n} z^{-1}}{1 - e^{-jW_0 n} z^{-1} - e^{jW_0 n} z^{-1} + z^{-2}} \right\}$$

$$= \frac{1}{2j} \left\{ \frac{2j \operatorname{sen}(W_0 n) z^{-1}}{1 - z^{-1} * 2 \cos W_0 n + z^{-2}} \right\} = \frac{z^{-1} \operatorname{sen}(W_0 n)}{1 - z^{-1} * 2 \cos(W_0 n) + z^{-2}}$$

Ahora con la propiedad Escalado en el Dominio Z

$$a^n x(n) \xleftrightarrow{Z} x(a^{-1} z)$$

$$Z\{a^n \operatorname{sen}(W_0 n)\} = \frac{a z^{-1} \operatorname{sen}(W_0 n)}{1 - 2a z^{-1} \cos(W_0 n) + a^2 z^{-2}}$$

$$R.O.C. = |z| > |a|$$

El exponente de a depende del exponente de z según la propiedad.

Ejemplo:

$$a^{-1} z$$

$$a^1 z^{-1}$$

$$a^2 z^{-2}$$

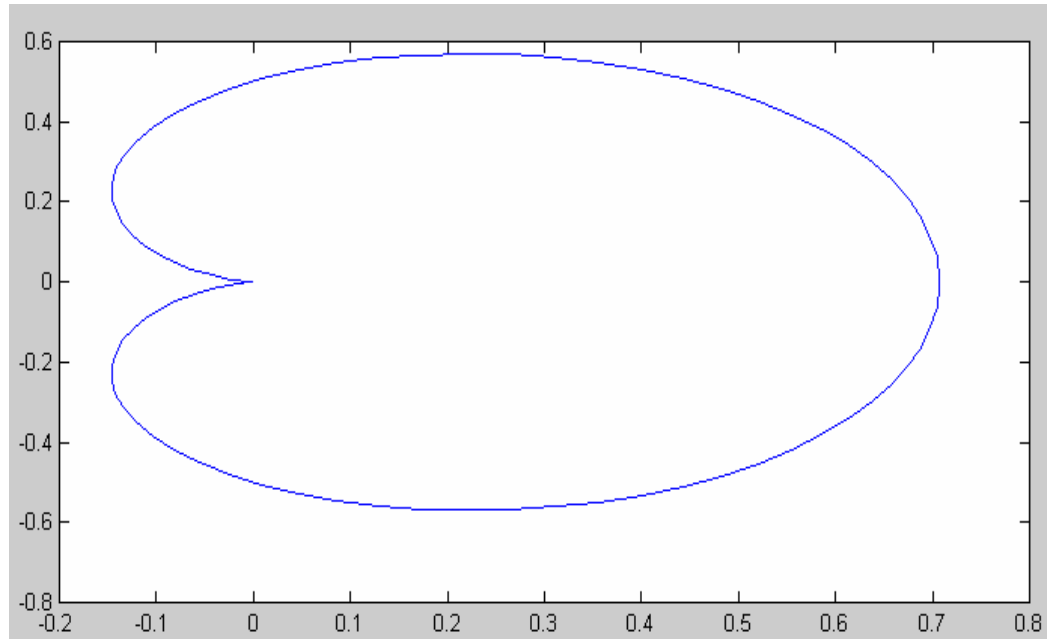


Fig. 3.4 grafica en Matlab, ejercicio de transformada Z

4. CONVOLUCIÓN

Calcular la convolución entre $x(n) = a^n \cup (n)$ y $h(n) = b^n \cup (n)$

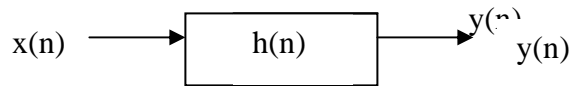
El problema matemáticamente se divide en:

Para $a \neq b$

$$y(n) = \sum_{K=-\infty}^{\infty} x(k) h^{(A)}(n-k) = \sum_{K=0}^n a^k \cup (k) b^{n-k} \cup (n-k) = \sum_{K=0}^n a^k b^{n-k} = b^n \sum_{K=0}^n \left(ab^{-1}\right)^{(D)}_k$$

$$y(n) = b^n \left(\frac{1 - \left(ab^{-1}\right)^{(E)}_{n+1}}{1 - ab^{-1}} \right) = \frac{b^{n+1} - a^{n+1}}{(F)_{b-a}}$$

(A). Para poder aplicar la convolución es preciso que el sistema sea lineal e invariante en el tiempo (LTI). De tal forma:



donde $X(n)$: señal de entrada

$h(n)$: respuesta del sistema (caja negra)

$y(n)$: señal de salida

(B). Reemplazando las señales del ejercicio. $X(n) = a^n \cup(n)$ y $h(n) = b^n \cup(n)$, como es invariante en el tiempo $h(n-k) = b^{n-k} \cup(n-k)$ los límites de las sumatorias se cambian $\sum_{K=0}^n$ ya que el sistema es causal.

(C). Por la cotación de los límites de la sumatoria podemos retirar sin ninguna implicación matemáticas los escalones unitarios $\rightarrow \cup(k)$ y $\cup(n-k)$.

(D). Por propiedades de la sumatoria: b^n es constante, puede salir de ella.

(E). Desarrollando la sumatoria tenemos.

(F). Simplificando.

Para $a = b$ tenemos

$$y(n) = \sum_{K=0}^n a^k a^{n-k} = a^n \sum_{K=0}^n 1 = (n+1)a$$

Este caso simplifica bastante el cálculo.

Ejemplo:

$X(n) = \{1, 2, 3, 1\}$ y la respuesta al impulso del sistema es $h(n) = \{1, 2, 1, -1\}$ el símbolo (_) significa el lugar de referencia, es decir para $n = 0$;

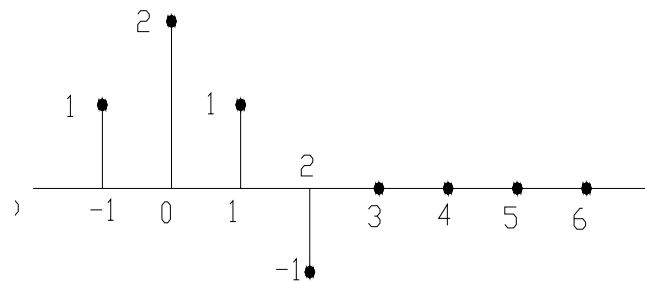
H(n-k)	Y(-1)	Y(0)	Y(1)	Y(2)	Y(3)	Y(4)	Y(5)	Y(6)
X(0)*h(n)	1	2	1	-1	0	0	0	0
X(1)*h(n-1)	0	2	4	2	-2	0	0	0
X(2)*h(n-2)	0	0	3	6	3	-3	0	0
X(3)*h(n-3)	0	0	0	1	2	1	-1	0
Y(n)	1	4	8	8	3	-2	-1	0



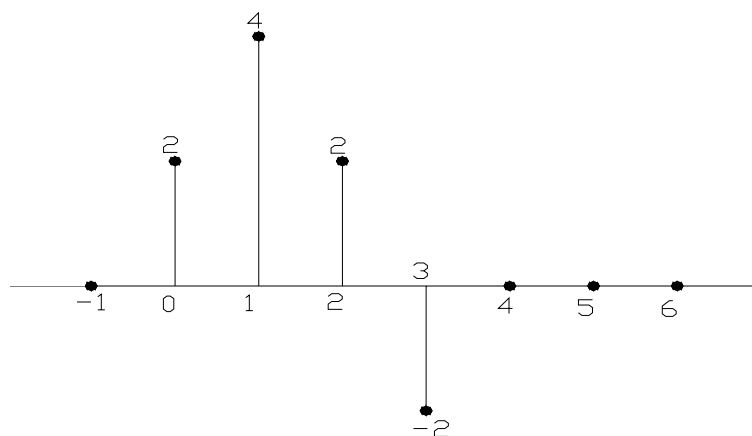
Observe que las cuatro primeras graficas tienen el mismo comportamiento, pero un factor $x(k)$ con $k = 0, 1, 2, 3$. y con un desplazamiento en el tiempo debido a $h(n-k)$.

Finalmente $y(n)$ se consigue de la sumatoria de estas señales o “ la sumatoria de las respuestas individuales a cada impulso de la señal de entrada “.

$K = 0, x(0)*h(n)$

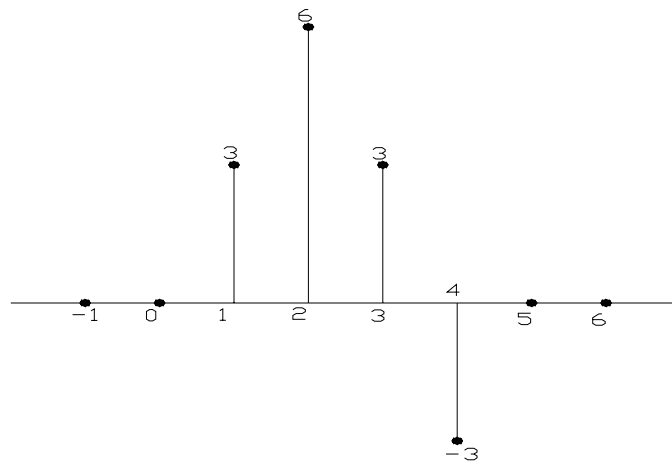


$K = 1, x(1)*h(n-1)$

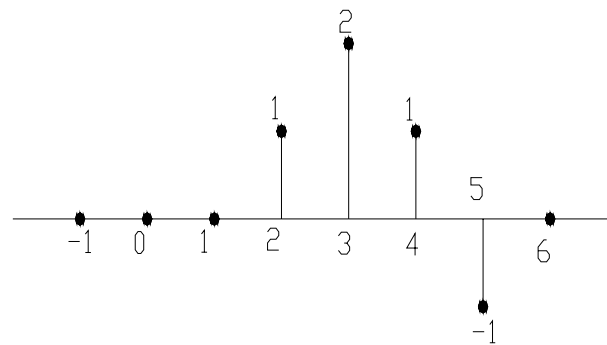




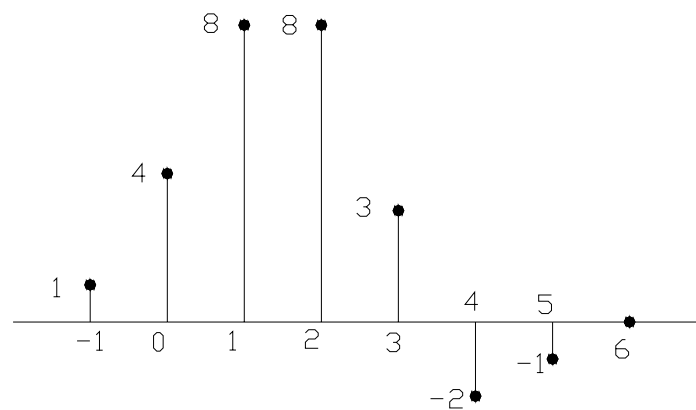
$$K = 2, \quad x(2) * h(n^2 - 2)$$



$$K = 3, \quad x(3) * h(n - 3)$$



$$y(n)$$



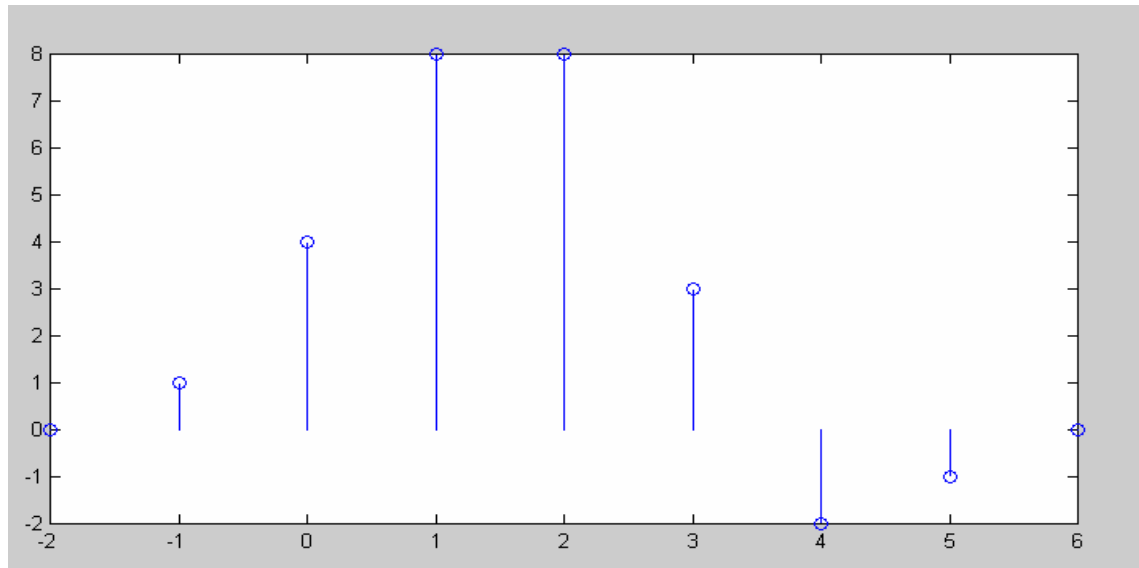


Fig. 3.5 grafica en Matlab, ejercicio de Convolución

5. FILTROS DIGITALES

5.1. Diseñar un filtro FIR pasabajo, con frecuencia de corte de 500 Hz para una señal muestreada a 8000 m/s.

La frecuencia de corte digital es:

$$\omega_c = 2\pi f_c = 2\pi \frac{500}{8000} = \frac{\pi}{8} = 0.3927 \text{ rad/muestra}$$

La respuesta al impulso, suponiendo 101 muestras y para hallar un filtro con ciertas simetrías respecto al impulso.

$$h(n) = \begin{cases} \frac{1}{8} & \text{para } n = 51 \\ \frac{\sin(\omega_c(n - 50))}{\pi n} & \text{para } n \neq 51 \end{cases}$$



5.2. Diseñar un filtro pasa bajos IIR con un ancho de banda de -3db en 500 Hz y una atenuación de -40db en 1000Hz.

f_c = Frecuencia de corte (-3db)

f_s = frecuencia especifica (-40db) “frecuencia de banda de rechazo”

$$f_c = 1000\pi = 2\pi f = 2\pi(500\text{Hz})$$

$$f_s = 2000\pi$$

Para cumplir con los -40db, $\delta_2 = 0.01$ entonces:

$$N = \frac{\log_{10}(10^4 - 1)}{2 \log_{10} 2} = 6.64$$

Es decir que tenemos N como el entero menor $\rightarrow N = 7$ y las posiciones de los polos son:

$$s_k = 1000 \pi e^{j(\pi/2 + (2k+1)\pi/14)}$$

$$k = 0, 1, 2, 3, \dots, 6$$

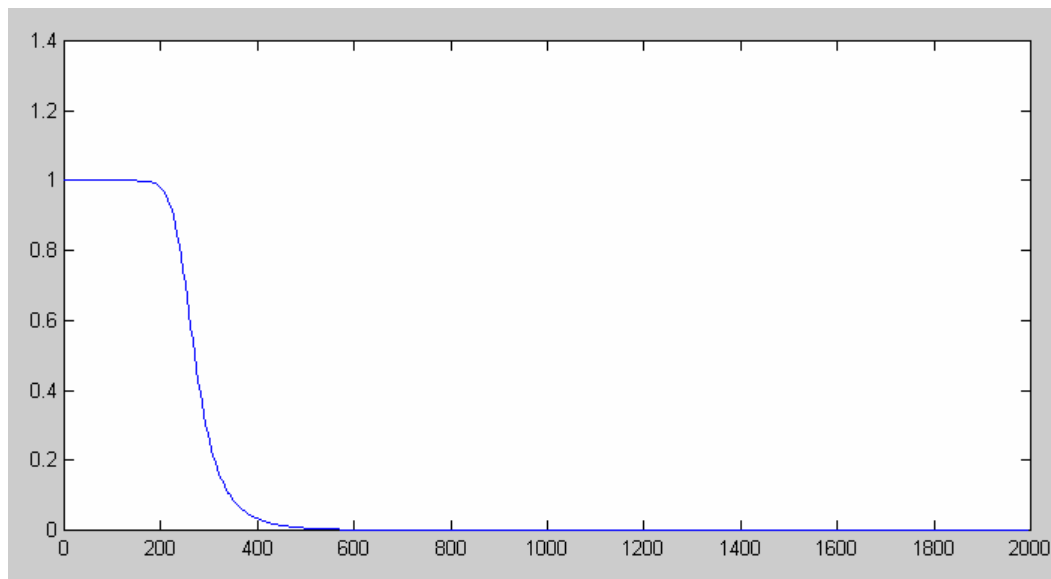


Fig. 3.6 grafica en Matlab, ejercicio de filtros digitales (5.2)



4. Diseño de GUI

4.1 Antecedentes

A lo largo del desarrollo de una Interfase Gráfica al Usuario (GUI), el cual incluye su creación e implementación; el diseño es parte fundamental en el proceso de creación. Al diseñar una GUI, se deben tener en cuenta todos los pasos previos a la generación del código que llevará al funcionamiento exitoso de la **Interfase Gráfica al Usuario**. Debido a que el diseño no siempre tiene un número específico de pasos a realizar, o un formato predeterminado, se pueden seguir muy variados caminos en el diseño de una GUI, pero existen muchos conceptos que se deben tener en cuenta para hacer que el trabajo sea más sencillo y productivo. El diseño de GUIs es un proceso básico que si es omitido antes de adentrarse a la programación, se podría perder gran cantidad de tiempo redundando en simples conceptos.

4.2 Principios del diseño

El diseño de GUIs no tiene mucho tiempo de haber sido implementado, puesto que anteriormente si se deseaba crear una *Interfase Gráfica* en MATLAB se tenía que seguir un proceso muy complicado de programación prácticamente manual y muy rudimentario, lo que hacía que la creación de GUIs fuera un terreno poco explorado por un usuario común y corriente. Esta reciente invención no es pretexto para intentar crear todo un nuevo método de diseño, ya que las cualidades universales de cualquier tipo de diseño se han mantenido intactas desde que se establecieron. Por lo que es confortante afrontar un problema de diseño de GUIs respaldado por siglos de conocimiento. A lo largo de la historia se han escrito gran cantidad de libros acerca del diseño, todos ellos coinciden en ciertos temas fundamentales, manteniendo el mismo patrón a través del tiempo, entre ellos:



Fig. 4.1 simplicidad, consistencia y familiaridad



Las ideas que son de nuestro interés, **simplicidad, consistencia y familiaridad**. Todas ellas son el centro representativo de muchas cualidades que se deben tener en cuenta al diseñar, pero indiscutiblemente la más importante de todas es la *simplicidad*, no necesariamente una GUI cargada de objetos y números contendrá toda la información que un usuario requiere, es así que al diseñar, se pueden identificar gran cantidad de objetos que no son necesarios. Para saber qué tan específicos estamos siendo en nuestros diseños podemos cuantificarlo cuestionando dos conceptos básicos, influidos por las ideas anteriores.

- a) ¿Cuánto tiempo le toma a un usuario realizar una tarea con la GUI la primera vez que la utiliza?
- b) ¿Cuánto tiempo le toma a un usuario realizar una tarea con la GUI una vez que le es familiar?

4.2.1 Simplicidad

Diseñando, la simplicidad es nuestra meta principal. Una GUI sencilla, tiene una apariencia limpia, y con sentido de unidad. Es muy fácil añadir objetos al construir una GUI, si ellos no aportan funcionalidad, o si lo hacen pero realmente esa funcionalidad sale sobrando, hay que quitarlos. Evitando aglomeraciones en la pantalla que confundan al usuario, es mejor sólo presentar las opciones que ayuden a desarrollar la función de la GUI.

Una vez que se ha identificado el primer objeto que se puede suprimir de nuestra GUI, será más sencillo encontrar muchos más que analizando el caso seguramente también saldrán sobrando.

4.2.1.1. Reducción del área de trabajo

Para qué utilizar varias ventanas si con una es suficiente para agrupar todos los objetos necesarios?

En el caso que se estén demostrando relaciones entre entrada y salida, no es necesario graficarlas en dos ventanas, si en una sola pueden colocarse ambas.

En la figura 4.2 ya se muestra una fusión de las ventanas minimizando el área de trabajo, haciendo una versión más amigable y funcional que es la finalidad primordial de éste trabajo de tesis. Ofreciendo funciones extras como: selección de la frecuencia, valor pico-pico, nivel DC, simetría, Botón de ayuda y una menú desplegable donde se pueden observar los diferentes temas para seleccionar.

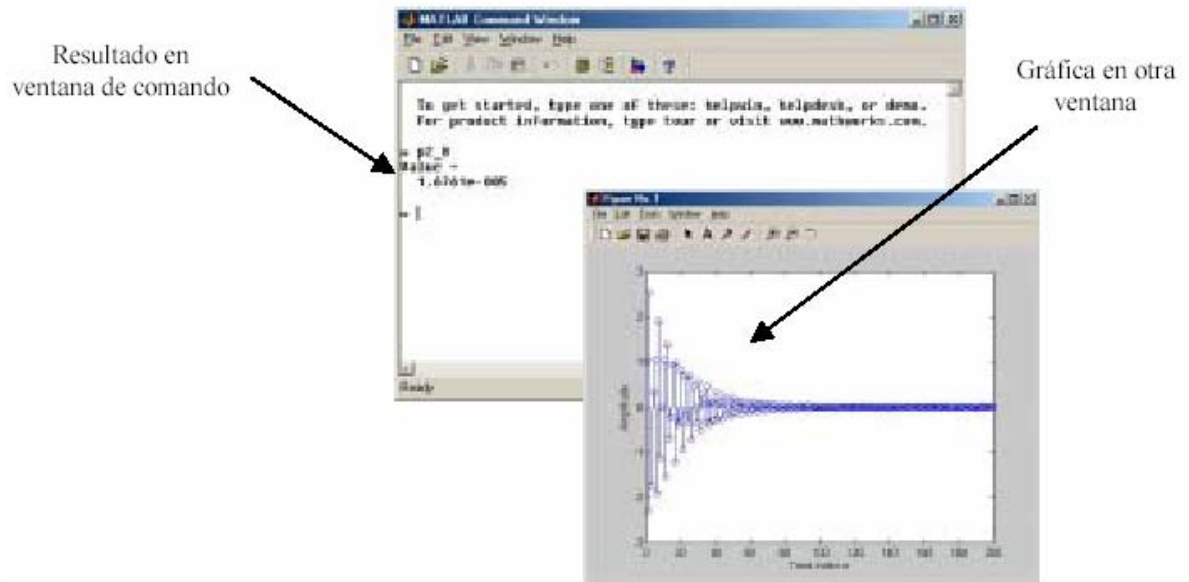


Fig. 4.2 Ventanas separadas

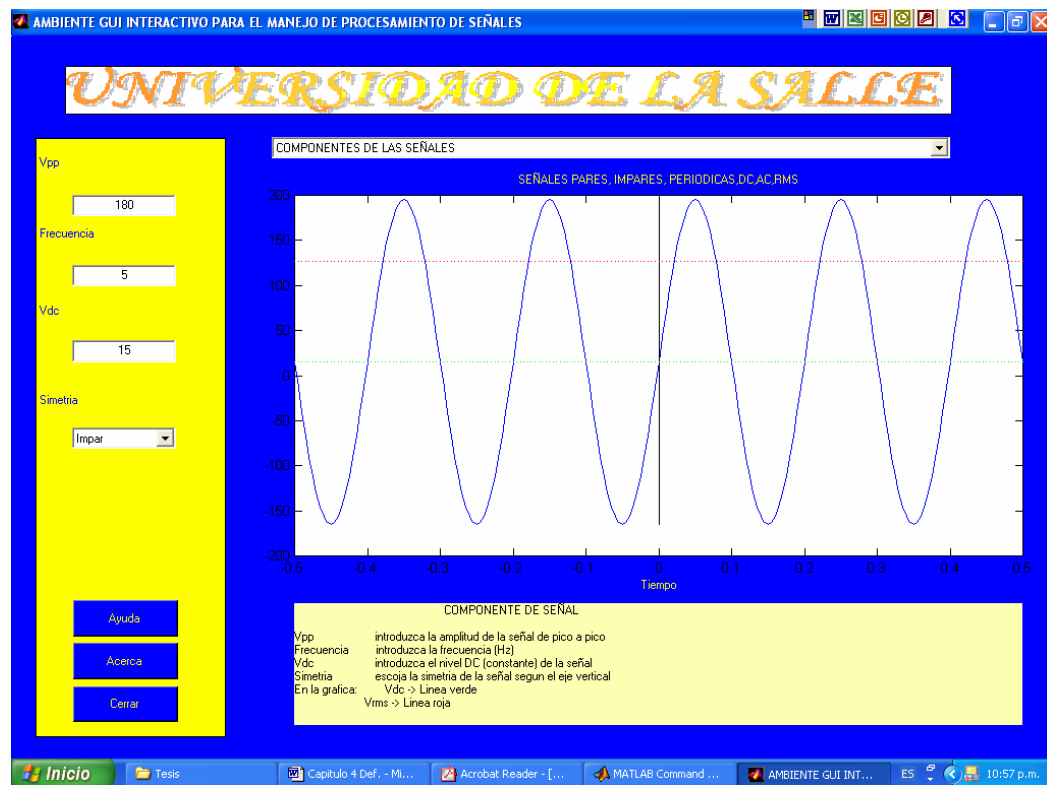


Fig. 4.3 Fusión en una sola ventana



4.2.2 Familiaridad

La consistencia ejemplificada en la figura anterior hace más sencillo moverse entre una GUI y otra, pues se crea el esperado ambiente de familiaridad entre ellas sin importar qué tan diferente es la función de cada una.

Si la GUI creada es en cierta manera familiar a los usuarios, será más agradable trabajar en ella y al utilizarla por primera vez se aprenderá a usarla más rápido. La familiaridad debe ser una buena pista que oriente a realizar una acertada suposición de los pasos a seguir e identificar la función de la GUI, sin necesidad de saber previamente de qué se trata, es decir sin conocimiento alguno de lo que se debe hacer para completar el trabajo.

Existen algunos ejemplos donde la experiencia de diseños externos sirve para poder añadirlos exitosamente a MATLAB mejorando la presentación y entendimiento de funciones. Por ejemplo utilizando algún programa de trazado de imágenes se diseña un fondo que sirve de base (**background**) de alguna GUI específica. MATLAB tiene la característica de poder cargar. Imágenes sobre ejes.

La familiaridad introduce a la gente a la GUI haciéndola sentir comfortable. Producido por la misma sensación de estar trabajando con objetos comunes aún siendo la primera vez que se utiliza la GUI.

4.3 Proceso de diseño

A pesar de lo mencionado que no existe un método predeterminado a seguir para el diseño de una GUI, existen sugerencias que son muy útiles para tener una idea de por dónde comenzar a trabajar. Como todos los procesos que tienen que ver con diseño, es de gran ayuda aprender a *saber pensar* en lugar de querer seguir paso a paso una receta existente. Es muy saludable diferenciar el proceso de diseño con el proceso de implementación de la GUI y pensarlos como dos fases diferentes que están relacionadas con el producto final pero que no influyen una con otra durante la creación. El corazón de este capítulo es recalcar en que se debe completar el diseño de la GUI antes de comenzar a implementarla, algunas veces no es posible terminar la parte de diseño totalmente pero se puede considerar llegar hasta donde más se pueda sobre papel. Si se comienza muy pronto la codificación, es decir la implementación del código de la GUI, la tendencia es llegar de una manera mucho más lenta a la conclusión del trabajo.

La fase de diseño comprende tres pasos los cuales son:

- Definición de la tarea



- Dibujar la GUI
- Prueba del diseño.

La fase de implementación:

- Escritura del código en MATLAB
- Prueba del código.

Desde luego que el diseño puede cambiar una vez comenzado el proceso de implementación y por supuesto existen muchas decisiones que no se pueden tomar previamente sino hasta que se haya comenzado a escribir parte del código y surjan interrogativas que nos harán regresar no sólo una vez sino recursivamente varias veces sobre la misma trayectoria **diseño - implementación** hasta que se logre el producto final esperado: una GUI funcionando correctamente.

Los mejores resultados finales se obtienen fruto de una visión idealizada inicial, el diseño ideal es el que existe en mente después de haber estado pensando sobre la tarea y los usuarios a los que esta enfocada la GUI. Este diseño es costoso, no tanto económicamente sino físicamente, pues requiere de una inversión sustancial de tiempo para desarrollar un concepto adecuado. Si se comienza con lo que se tiene a la mano, el resultado carecerá de valor; pero si se comienza con un ideal, es sorprendente cómo se puede sacar provecho de la tecnología con la que contamos actualmente haciendo el trabajo más sencillo y mejor presentado.

Si se hace un dibujo de la GUI y se prueba su función sobre papel, será más sencillo identificar que es lo que se quiere expresar al usuario, teniendo una perspectiva desde el mundo externo, sintiendo, observando, probando, no quedándonos solo con el papel de diseñadores. Trabajando sobre papel como no hay un código involucrado resulta hasta divertido hacer la labor de la computadora, identificando así si la GUI hace lo que se tiene pensado o lo que el usuario requiere.

Al momento de estar listo para escribir el código ya se debe haber hecho un diagrama o un conjunto de diagramas de la GUI (trazos), y una descripción completa de la funcionalidad asociada a cada objeto de la GUI (*callbacks*).

4.3.1 Definición de funciones y trazado de la GUI

Después de tener la idea o ideas de lo que llegará a ser una GUI aunque se encuentren vagas en cierta manera, comenzar a plasmar esas ideas sobre papel es de gran ayuda, aunque no se sepa exactamente en dónde irán posicionados ni exactamente cuantos objetos formarán parte de la GUI.

Como ejemplo a continuación se detalla la concepción de la GUI del **Ambiente GUI**. A grandes rasgos debe mostrar el menú donde se encuentran los botones de comando y entrada de los datos, los ejes donde se generara la grafica y un texto instructivo de cómo ingresar los datos a la interfase.

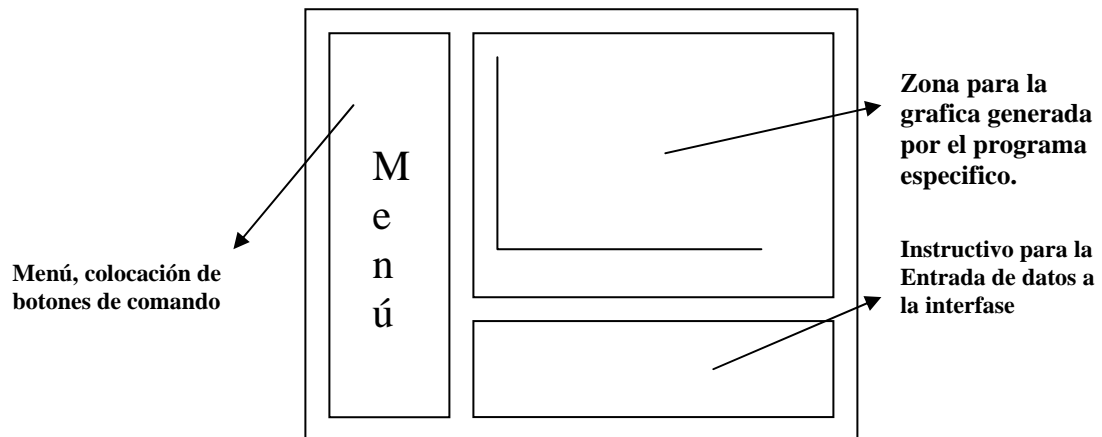


Fig. 4.4 Bosquejo inicial

En un segundo bosquejo se realiza la afinación de detalles determinando las funciones que tendrá la GUI, las cuales son: los cuadros de edición donde se introducen los diferentes datos. Además de tener botones adicionales los cuales sirvan para cerrar el programa y buscar las ayudas de los comandos con lo que trabaja la GUI de un tema específico y por último un botón para entrar a una ventana donde se va a dar una referencia del título de la tesis de grado, autor, director de tesis, institución y facultad donde se desarrollo esta tesis y el año de su creación. En este punto después de haber imaginado la situación de los elementos en diferentes lugares y determinado cual sería su posición definitiva, nos encontramos en el último esfuerzo en cuanto a diseño se refiere.

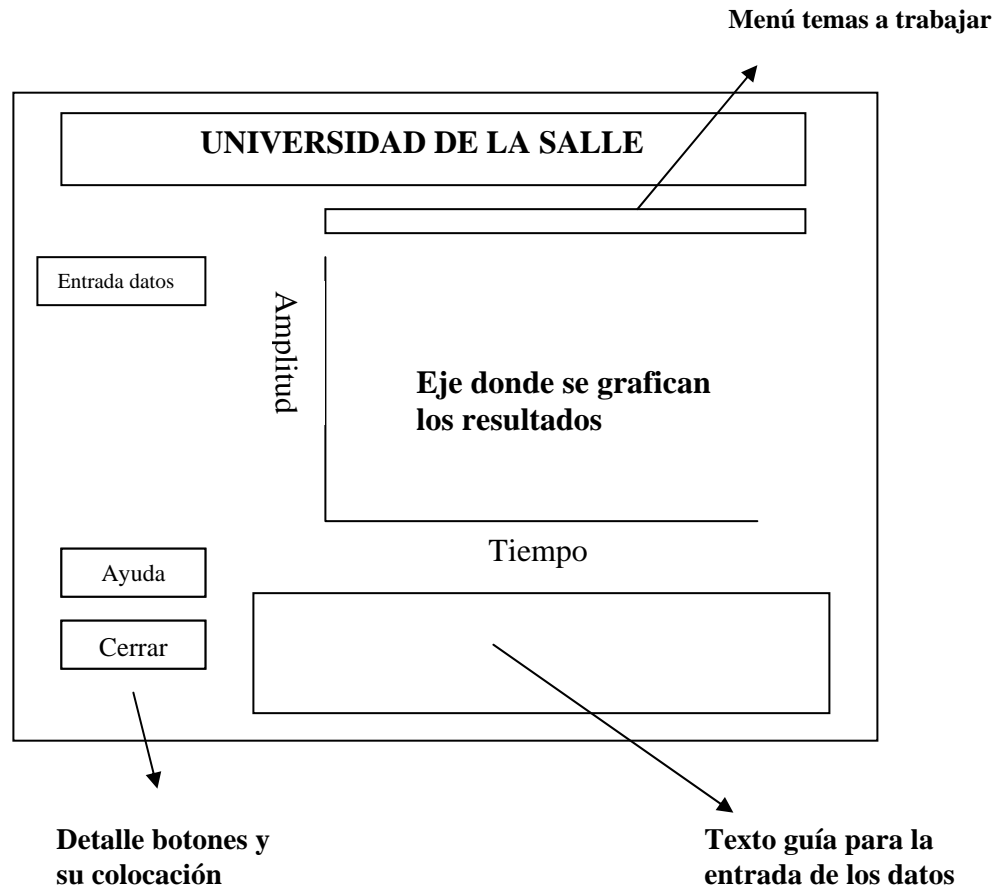


Fig. 4.5 Segundo bosquejo



5. Implementación de GUI

5.1 Antecedentes

En este punto del trabajo se asume que ya se tiene una idea firme de lo que se intenta construir, ahora hay que aprender cómo construirlo. Basados en *guide* para implementar el código en MATLAB. Dividiendo el material en dos partes, cómo usar *guide* para trazar una GUI, y cómo escribir las llamadas de función (**callbacks**), que harán el trabajo detrás de pantalla (memoria).

Cualquier cosa que aparece en una figura de MATLAB, tiene un identificador único no importando el objeto, carácter que permite en cualquier momento poder identificarlo y modificarlo cualquiera que sea. Aunque esos identificadores únicos de manejo siempre se encuentran presentes en componentes de las figuras de MATLAB, regularmente no hay porqué preocuparse, puesto que tienen asignados valores predeterminados en los cuales no influye el usuario. Además del identificador, los objetos tienen diferentes propiedades; en caso que se deseen variar, para añadir valor agregado, es posible como característica adicional para mejorar la presentación y entendimiento de la figura.

5.2 Editor de propiedades

El editor de propiedades, la herramienta de *guide* de mayor uso, es muy útil para navegar fácilmente entre las diferentes categorías de los tipos de objetos; desde la raíz, la figura, controles, ejes, y sus respectivas características. Ya sea para adquirir o establecer alguna propiedad de cualquiera de los objetos mencionados.

Una ventana de figura y sus componentes tienen un cierto número de propiedades independientes, como: nombre, visibilidad, color, etc. Para ver y modificar éstas propiedades desde la *línea de comando* se necesitan utilizar los comandos *set* y *get* para manejar la imagen; en conjunto con el comando *.gcf*. (**get current figure**).

Teclear solo el comando *gcf* no tiene mucha razón, debemos tener presente que obtendremos como resultado el número de figuras activas (ej: **ans = 2**).

Si queremos variar alguna característica y son varias ventanas las que se encuentran abiertas, el comando surtirá efecto en la última que fue abierta o seleccionada.

Trabajar desde la ventana de comando para variar características es muy complicado, debido a que los objetos tienen solo algunas propiedades en común y muchas más son propias de cada objeto, se requiere ya sea una memoria

privilegiada para recordar todos los nombres exactos de cada una de las funciones y su sintaxis, o estar consultando constantemente la ayuda de MATLAB que lo hace un trabajo muy pesado. En cambio el apreciado apoyo del **editor de propiedades** permite revisar los identificadores de los objetos componentes de la figura de acuerdo a su jerarquía.

El editor de propiedades tiene dos modalidades de visualización las cuales comparten una parte proporcional de la ventana. En la tercera parte se encuentran los controles que activan y desactivan esta subdivisión, botón de ayuda, botón de cerrar y por último una sección de información donde se despliega una breve explicación cada vez que se selecciona alguna característica de la lista de propiedades.

Una gran ventaja del editor de propiedades es que tiene funciones inteligentes, por ejemplo si tecleamos en el cuadro de propiedades alguna letra, digamos .S., automáticamente la lista se posiciona sobre todas las propiedades que comienzan con esa letra o frase en dado caso.

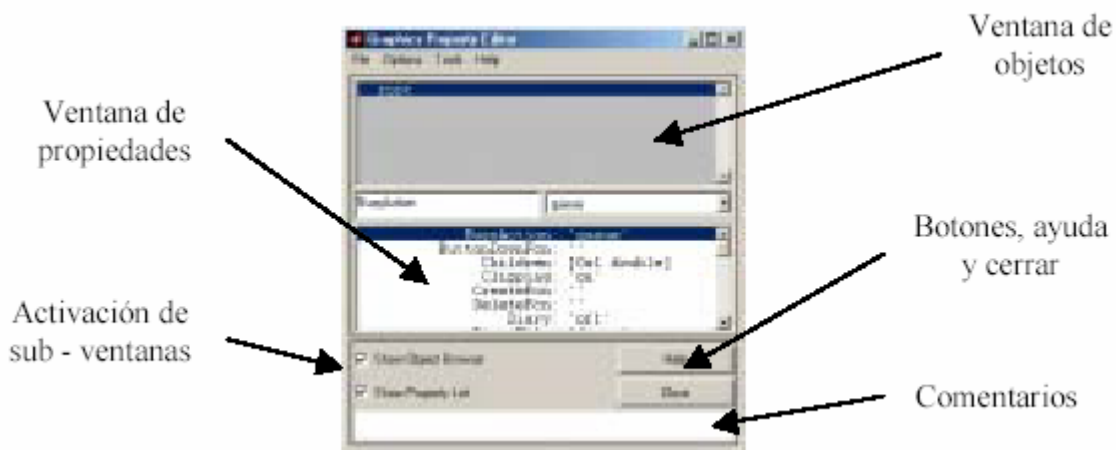


Fig. 5.1 Editor de propiedades

Algunas propiedades como .Unidades. (**Units**) tienen pocas posibilidades de elección, es así que el campo de valor se sustituye con un **botón de aparición** que permite seleccionar únicamente de entre las opciones ofrecidas y no da margen a que el programador acceda valores propios, éste es un proceso muy



conveniente pues facilita la selección de características. En el ejemplo proporcionado: .Unidades., las opciones son seis entre ellas: pulgadas, centímetros, puntos, píxeles, normalizado, y caracteres. En pocas palabras por medio de ésta acción, el editor de propiedades nos anticipa las opciones posibles.

El editor de propiedades es la *varita mágica* en cuanto a gráficos se refiere en MATLAB. Para sacarle el mayor provecho al emplearlo se pueden usar los siguientes secretos.

Se pueden seleccionar múltiples objetos y variar características que tengan en común en una sola partida. Así si nos interesa ponerle el mismo color a tres objetos se seleccionan al mismo tiempo (dejando presionado el botón *.shift.*, al seleccionar uno y otro) y en el campo de **.Background Color**. Se ajusta el deseado.

Como ya se había mencionado, el editor de propiedades es .inteligente., pues al escribir en el campo de propiedad una letra o una pequeña serie y picando la tecla de tabulador, automáticamente se posiciona la lista de objetos en las propiedades que comienzan con esa letra o letras. Brindando la comodidad de no estar pensando en el nombre exacto de la propiedad que se desea variar, ahorrando tiempo en intentar teclear el nombre completo aunque lo supiéramos.

Fraccionamiento después del primer espacio: en el campo de propiedad se ignora cualquier cosa escrita después de un espacio, lo que también ahorra gran cantidad de tiempo. Pues podemos escribir una propiedad diferente, prácticamente sobre la anterior, sin tener que borrar el nombre.

5.3 Panel de control

El panel de control es la ventana que concentra el acceso a las herramientas de *guide*, es por ello que se le puede nombrar ventana de exhibición. Además que permite añadir a una figura controlada elementos que se detallan a continuación. En la ventana del panel de control cuando la figura que se esta trabajando está activa, los controles de adición aparecen inactivos.

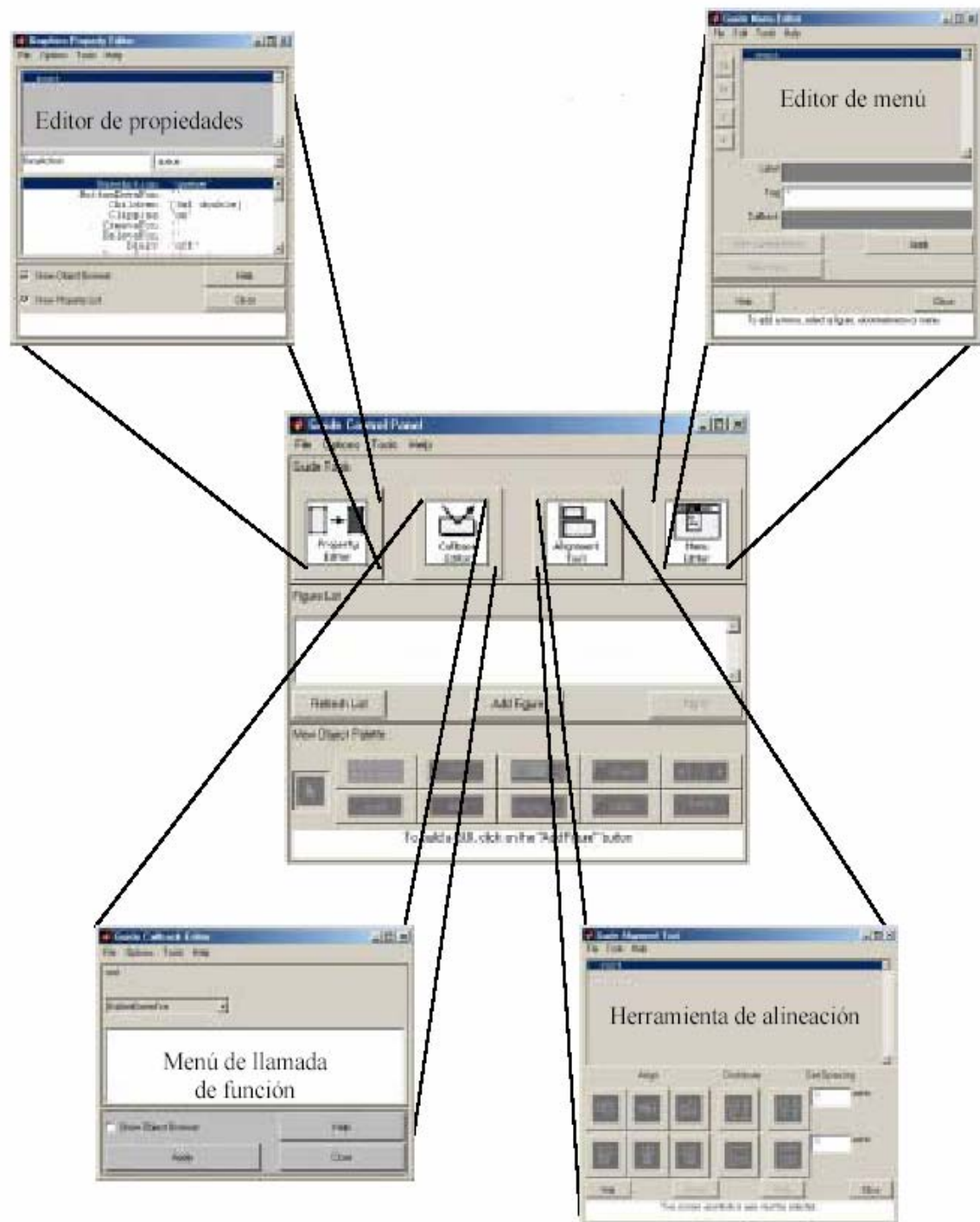


Fig. 5.2 Herramientas de GUIDE



El panel de control se divide en tres partes principales:

- Herramientas de *guide* (arriba).
- Lista de figuras controladas (centro).
- Barra de objetos nuevos (abajo).

Las herramientas de *guide* incluyendo al panel de control son: editor de propiedades, editor de llamadas de función, herramienta de alineación y editor de menú. Cuando alguno de ellos se encuentra abierto, en el panel de control se visualiza presionado el botón que hace referencia al aparejo.

En la parte inferior del panel de control se encuentra la barra de nuevos elementos. Para añadir un objeto como un botón de presión (*push button*), se selecciona el objeto en el panel de control, y sobre la figura **controlada** se traza al tamaño deseado (característica que puede ser variada posteriormente). Si se requieren más objetos iguales, es posible mientras está seleccionado copiarlo y pegarlo (**copy - paste**) sobre la misma figura, arrastrándolo con el *mouse* para colocarlo en la posición deseada.

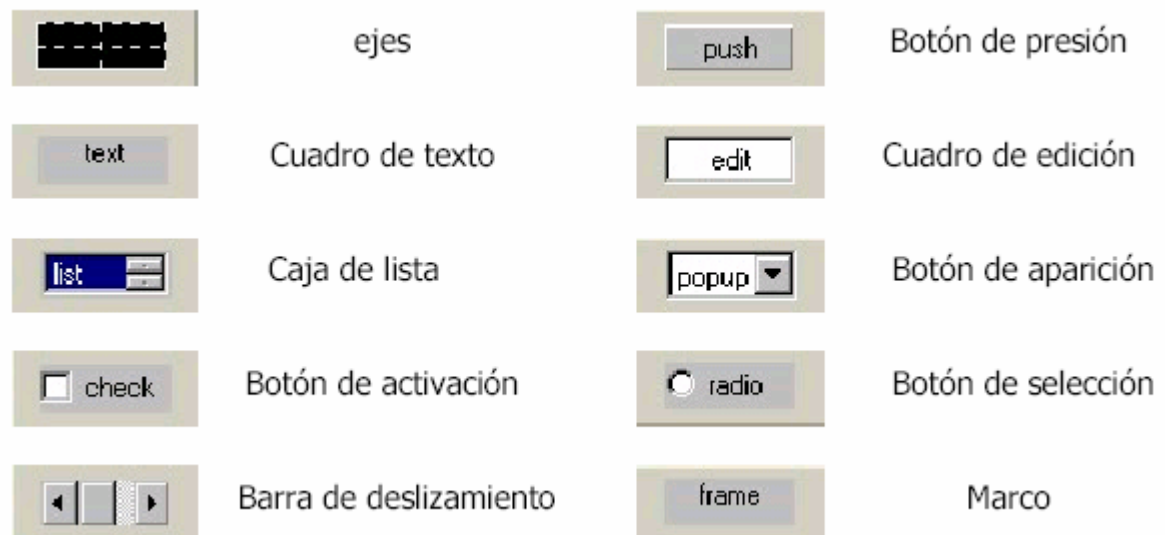


Fig. 5.3 Objetos disponibles desde el panel de control

Ahora para variar las características de cada objeto, como ponerle nombre, nos apoyamos del editor de propiedades donde de acuerdo a la jerarquía correspondiente hay que buscar dentro de raíz-figura-botón de presión #1. La propiedad a variar es *.String.*, y el nombre se coloca entre comillas simples para que MATLAB identifique que se trata de una cadena (ASCII) y no una variable.

A la mitad del panel está la lista de figuras que se localizan bajo el dominio de *guide*, ahí se muestran las figuras abiertas y su estado (activas o controladas).

Cuando una figura se encuentra controlada, se pueden mover con libertad los ejes, botones y cualquier otro objeto que esté en la figura simplemente con seleccionarlo con el *mouse* y arrastrarlo. La figura que se halla controlada, muestra sobre su perímetro una semicuada de líneas blancas que indican que se encuentra en modo de edición.

Una vez que todo está en orden y como es deseado, inclusive después de haber trabajado con el editor de llamadas de función, ya se puede **activar** la figura para habilitar los componentes quedando listos para desempeñar la función que les corresponde respectivamente.



Fig. 5.4 Activación de figura

Cuando se escribe la instrucción *guide* directamente de la ventana de comando, si hay una figura abierta, automáticamente se pone en modo controlada, en caso contrario se abre una nueva figura a la par del panel de control.

Ahora para activar la gráfica, como se muestra en la figura 5.4, se selecciona la palabra **“Controlled Figure #1”** en la lista de figuras, con la acción

automáticamente cambia a “**Active Figure #1**”, para confirmar se da clic sobre el botón **Aplicar**. De inmediato la figura está lista para comenzar a usarla.

Se pueden controlar varias figuras a la vez, pero se torna un tanto caótico, pues sólo se puede abrir una ventana de cada herramienta de *guide*, entonces para establecer características de una u otra figura, se debe estar compartiendo la ventana de la herramienta en cuestión, que en algunos casos causa confusiones y hace que se pierda tiempo, por ello es muy recomendable modificar una sola figura a la vez.

5.4 Editor de llamadas de función

Cuando los objetos se han posicionado, es tiempo de determinar qué función realizará el objeto cada vez que sea ejecutado. Es ahora trabajo del editor de llamadas de función ayudarnos a establecer la tarea que realizará cada elemento. De igual manera que el editor de propiedades cuenta con tres secciones divididas en sub-ventana de objetos, sub-ventana de **llamadas de función** (comandos a ejecutar cuando se ejecute el objeto), y la última región donde están los botones de ayuda, cerrar, aplicar función y control de activación de ventana de objetos.

Al momento de escribir los comandos en la sub-ventana no es necesario colocarlos entre comillas, pues perderían su sentido de función. También se debe tener claro que no hay limitante en el número de comandos a efectuar, no importando qué tan simples o complejos sean.

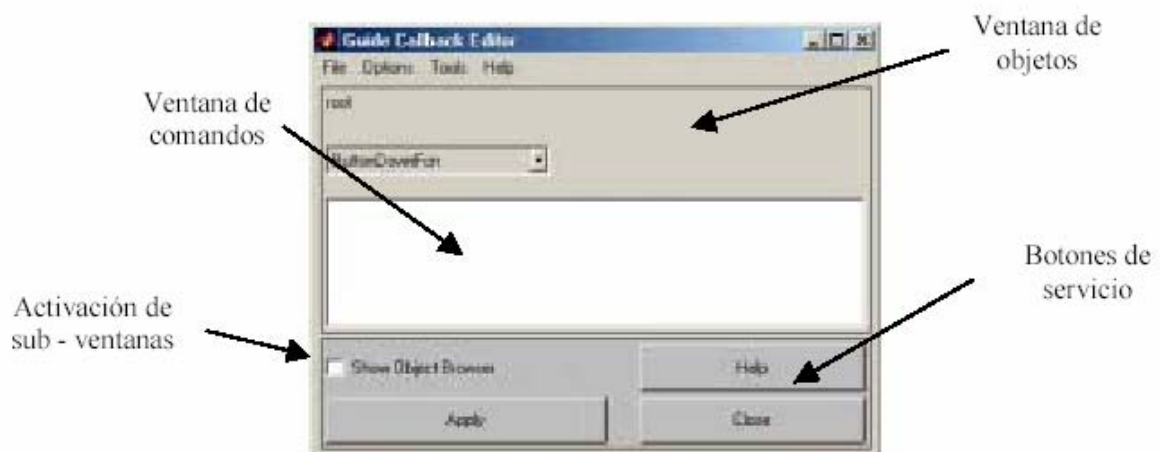


Fig. 5.5 Editor de llamadas de función

5.5 Herramienta de alineación

Es muy útil para poner en orden los objetos de una figura. La ventana que aloja a la herramienta de alineación tiene botones que permiten ya sea alinear o distribuir. Alinear con respecto a una posición: izquierda, derecha, abajo, arriba, centrado (vertical- y horizontalmente). Distribuir con respecto a un objeto seleccionado un cierto espacio en puntos entre otro u otros objetos seleccionados.



Fig. 5.6 Herramientas de alineación

5.6 Editor de menú

Se pueden agregar menús personalizados complementando los que aparecen en la parte superior de la figura, que son: archivo, edición, herramientas, ventana, ayuda. Mediante el editor de menú es imposible hacer cualquier tipo de modificación a los ya mencionados, pero se pueden añadir y editar nuevos, disponibles una vez que la figura haya sido activada.

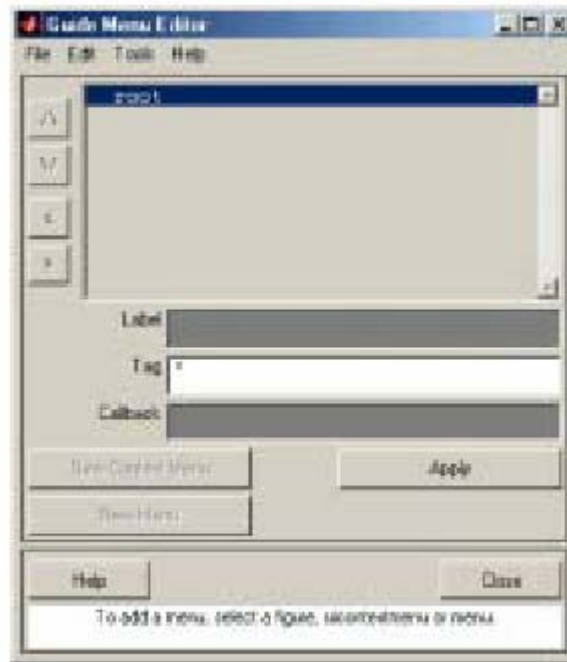


Fig. 5.7 Editor de menú



6. COMPONENTES DEL AMBIENTE GUI

La interfase grafica de usuario fue el resultado de horas de delicado y minucioso trabajo para llegar a cumplir con los requerimientos de esta tesis de grado, por esta razón y como complemento se desarrollo una serie de practicas para la comprobación de dicha interfase (ver apéndice A).

6.1 Características

El Ambiente GUI Interactivo está integrado por 2 ventanas incluyendo la de presentación, en la cual el usuario tiene conexión y acceso a las demás GUIs. El concepto esta desarrollado de la siguiente manera, las GUIs están agrupadas en capítulos de acuerdo al contenido de información que cada una representa.

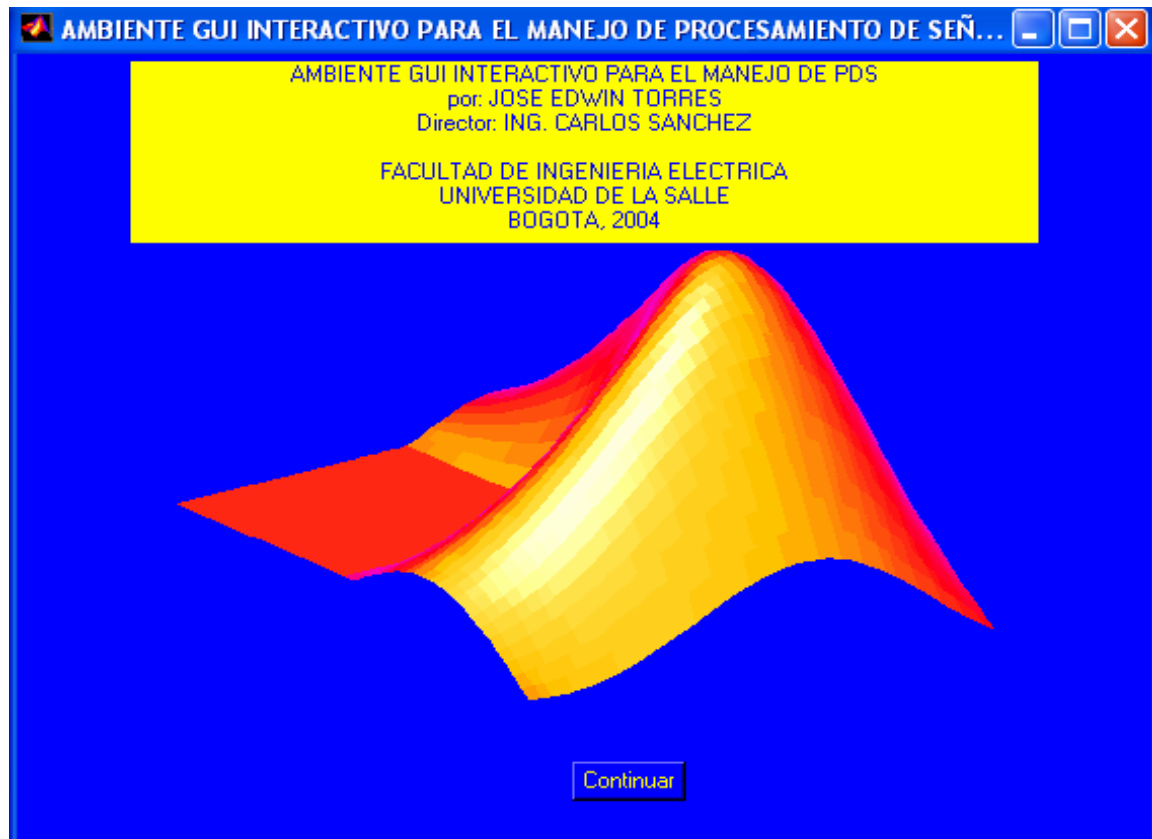


Fig. 6.1 Ventana de presentación



6.2 DETALLES Y EXPLICACION DE LAS GUIs

6.2.1. COMPONENTES DE LAS SEÑALES

En esta GUI se genera una señal sinusoidal en la que se puede establecer la frecuencia, amplitud y simetría. Además de obtener sus niveles DC y AC (Vrms).

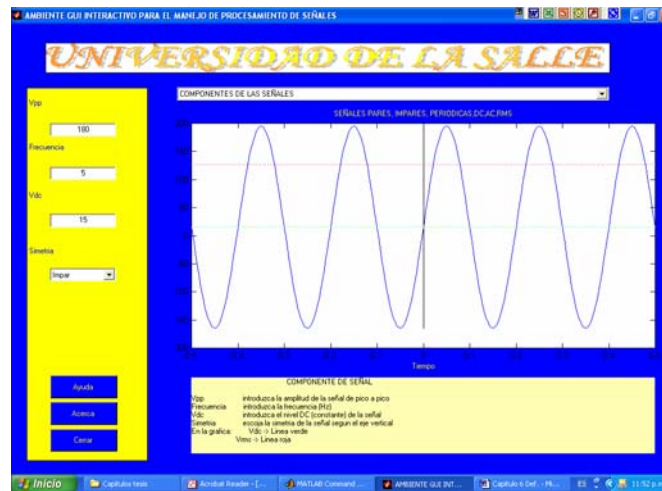


Fig. 6.2 GUI componentes de las señales

6.2.2. CONVOLUCION

Esta GUI grafica la salida de un sistema, a partir de la secuencia de la señal de entrada $x(n)$ y el comportamiento al impulso de dicho sistema $h(n)$. Tiene la funcionalidad de cambiar el punto de inicio de cualquiera de las dos secuencias.

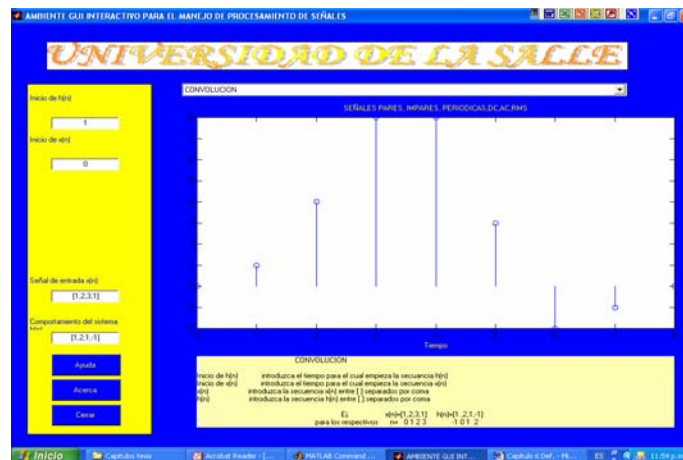


Fig. 6.3 GUI convolución



6.2.3 TRANSFORMADA DE FOURIER CONTINÚA

Esta GUI trabaja a partir de la secuencia de una señal de entrada y grafica el espectro de la señal además de la misma señal por medio de la transformada de fourier.



Fig. 6.4 GUI transformada de fourier continúa

6.2.4. TRANSFORMADA DE FOURIER DISCRETA

Esta GUI trabaja a partir de la secuencia de una señal de entrada y grafica el espectro de la señal además de la misma señal por medio de la transformada de fourier. Grafica el numero de N espectros.

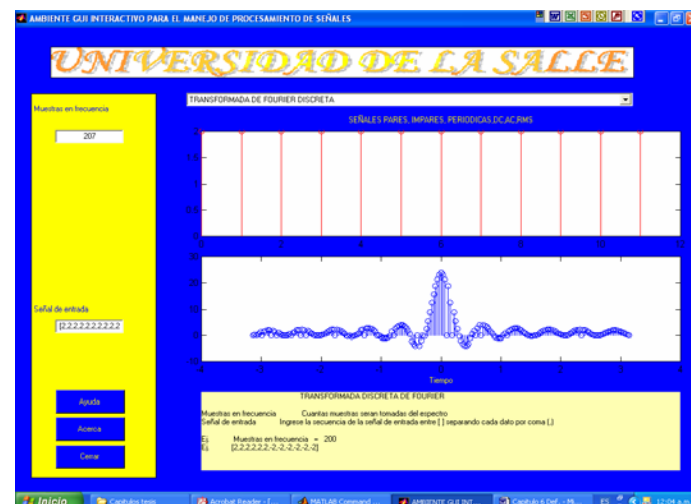


Fig. 6.5 GUI transformada de fourier discreta



6.2.5. TRANSFORMADA DE LAPLACE

Esta GUI grafica la traza de bode en un sistema.



Fig. 6.6 GUI transformada de laplace

6.2.6. TRANSFORMADA Z

Esta GUI grafica el diagrama de Nyquist de una funcion de transferencia Z, en el plano Z.

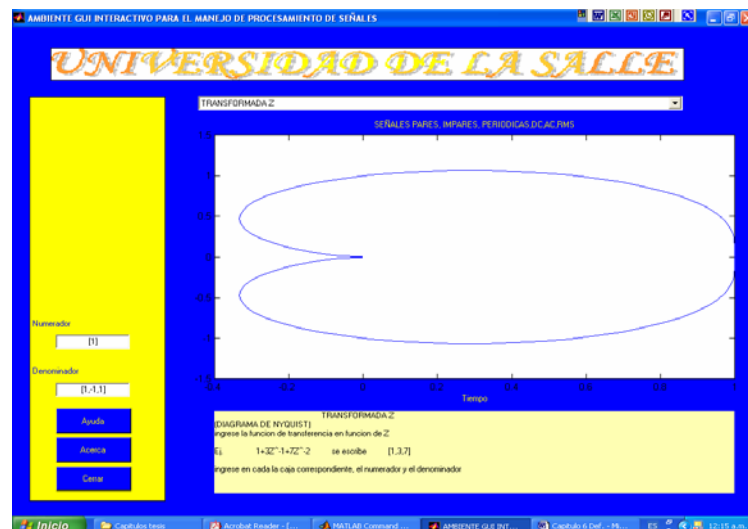


Fig. 6.7 GUI transformada Z

6.2.7. FILTROS DIGITALES

Esta GUI grafica el comportamiento de un filtro pasa bajos teniendo en cuenta el orden del filtro, la frecuencia de corte y la frecuencia de muestreo de la señal. Además lista los coeficientes de un filtro Butterworth.

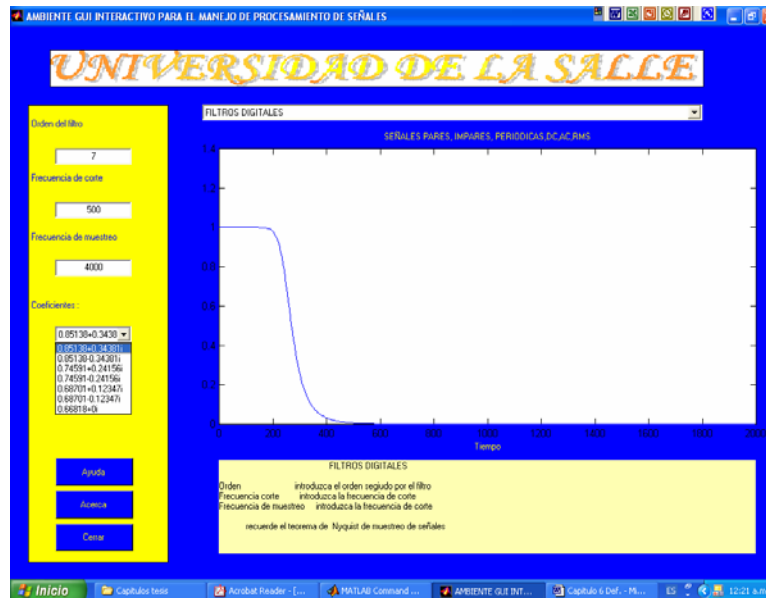


Fig. 6.8 GUI filtros digitales



CONCLUSIONES

La consistencia y simplicidad de un sistema es una de las bases del éxito de tal, puesto que lo hace llamativo y de gran alcance ante los usuarios de este.

La familiaridad y guianza de una interfase en ambientes de aprendizaje e investigación fomentan correctos hábitos de estudio, enfocando la atención del usuario directamente al objeto de estudio y no al aprendizaje del funcionamiento de la plataforma.

El emergente avance de la tecnología basada en PDS exige cada vez mas el conocimiento de esta área en asignaturas de pregrado, para las cuales hace falta herramientas de apoyo básicas y funcionales.

El estudio guiado de un área tan importante como el procesado digital de señales (PDS) dentro de una herramienta computacional como MATLAB genera precisión en los resultados y seguridad al estudiante.

El aprendizaje de nuevas tecnologías tanto teóricas como practicas son mas efectivas con la introducción de ambientes computacionales gráficos e interactivos.

La versión utilizada de MATLAB sobre la cual se trabajo es la 5.3 (edición 11).

La posibilidad de utilizar la doceava edición de MATLAB la tuve cuando el trabajo se encontraba cerca de la mitad es por eso que tome la decisión de continuar usando la versión con la que había comenzado el proyecto.



BIBLIOGRAFÍA

1. S.K. Mitra. Laboratorio de procesamiento digital de señales. McGraw-Hill. Nueva York, 1.999.
2. García de Jalón, Javier. Aprenda Matlab 5.3 como si estuviera en primero. Universidad de Navarra, San Sebastián, Agosto, 1.999.
3. Ogata, Katsuhiko. Ingeniería de control moderna. Tercera edición. Prentice Hall. México.
4. The MathWorks. Matlab Help Desk. Version 5.3 (R11), January 21 de 1.999.
5. The MathWorks. Matlab Help Windows. Version 5.3 (R11), January 21 de 1.999.
6. Internet. The MathWorks.com
7. <http://mec21.etsii.upm.es/ayudainf/aprendainf/Matlab53/matlab53.pdf>
8. <http://ccc.inaoep.mx/~ralfaro/Tutoriales/Matematicas/matlab53.pdf>
9. <http://www.terra.es/personal4/banky2/manual.html>



APÉNDICE A

PRACTICAS DE PDS UTILIZANDO EL AMBIENTE GUI INTERACTIVO

UNIVERSIDAD DE LA SALLE – FACULTAD DE INGENIERIA ELECTRICA.

PRACTICA No. 1: COMPONENTES DE LAS SEÑALES.

Trabajando con la **GUI** de componentes de las señales del **ambiente GUI interactivo** determinar las diferentes variaciones de la grafica pre-establecida (sinusoidal) al variar sus componentes de valor pico-pico, frecuencia y nivel DC.

a. frecuencia: 20, Vpp: 60, Nivel DC: 45

1. Determine el nivel RMS y DC de la grafica.
2. Que sucede si se la cambia la simetría a la señal.
3. Como afecta la frecuencia la señal.

b. Vpp: -100, Nivel DC: 10, Frecuencia: 5

1. Determine los niveles RMS y DC de la grafica. Explicarlos
2. Que influencia tiene la frecuencia en una señal.
3. Para que sirve el Valor pico-pico en una señal.
4. Que sucede si se varia el nivel DC a 120

Explique las observaciones vistas en los puntos a y b y dibuje las graficas correspondientes.

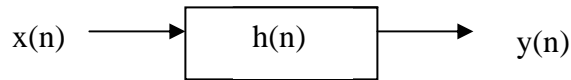
UNIVERSIDAD DE LA SALLE – FACULTAD DE INGENIERIA ELECTRICA.

PRACTICA No. 2: CONVOLUCIÓN.

Trabajando con la **GUI** de convolución del **ambiente GUI interactivo** determinar la señal de salida de un sistema lineal e invariante en el tiempo (LTI).



$X(n)$ = señal de entrada
 $h(n)$ = respuesta del sistema (caja negra)
 $y(n)$ = señal de salida



Introducir la señal de entrada $x(n) = [2, -8, 5, 10, 0.1]$

Introducir la señal de la caja negra $h(n) = [5, -0.6, 1]$

Inicio en $X(n) = -8$

Inicio en $h(n) = 1$

Observar la señal de salida $y(n)$ y explicarla

UNIVERSIDAD DE LA SALLE – FACULTAD DE INGENIERIA ELECTRICA.

PRACTICA No. 3: TRANSFORMADA DE FOURIER CONTINUA.

Trabajando con la **GUI** de transformada de fourier continua del **ambiente GUI interactivo** determinar la transformada de fourier continua de la señal de entrada.

a. Señal de entrada = $[0.5, -0.6, 8, -4, 6]$

1. Observar y compara el espectro y la señal grafica de salida.
2. Dibujar y explicar los tipos de señales

b. Señal de entrada = $[-1, 1, -10, -5]$

1. Observar y compara el espectro y la señal grafica de salida.
2. dibujar y explicar los tipos de señales



UNIVERSIDAD DE LA SALLE – FACULTAD DE INGENIERIA ELECTRICA.

PRACTICA No. 4: TRANSFORMADA DE FOURIER DISCRETA.

Trabajando con la **GUI** de transformada de fourier discreta del **ambiente GUI interactivo** determinar la transformada de fourier continua de la señal de entrada estableciendo el numero de muestras.

a. Señal de entrada = [0.5,-0.8,15,3,-6]

Numero de muestras = 50

1. Observar y compara el espectro y la señal grafica de salida.
2. Dibujar y explicar los tipos de señales

b. Señal de entrada = [-10,0.65,-0.15,-5]

Numero de muestras = 20

1. Observar y compara el espectro y la señal grafica de salida.
2. dibujar y explicar los tipos de señales

UNIVERSIDAD DE LA SALLE – FACULTAD DE INGENIERIA ELECTRICA.

PRACTICA No. 5: TRANSFORMADA DE LAPLACE.

Trabajando con la **GUI** de transformada de laplace del **ambiente GUI interactivo** determinar la traza de bode de la función de transferencia en términos de S.

Numerador = [5,-05]

Denominador = [45,-0.56,10]

Dibujar y observar la grafica de la traza de bode y determinar que funcionamiento tiene.



UNIVERSIDAD DE LA SALLE – FACULTAD DE INGENIERIA ELECTRICA.

PRACTICA No. 6: TRANSFORMADA Z.

Trabajando con la **GUI** de transformada Z del **ambiente GUI interactivo** determinar el diagrama de Nyquist de una función de transferencia en función de Z.

Numerador = [-5,.05,6]

Denominador = [12,-0.05,4,-8]

Observar la grafica en el plano Z y compárela con otra función de transferencia.

UNIVERSIDAD DE LA SALLE – FACULTAD DE INGENIERIA ELECTRICA.

PRACTICA No. 7: FILTROS DIGITALES.

Trabajando con la **GUI** de filtros digitales del **ambiente GUI interactivo** determinar la grafica del filtro pasa bajos teniendo en cuenta el orden del filtro, la frecuencia de corte y la frecuencia de muestreo.

Orden del filtro = 10

Frecuencia de corte = 60

Frecuencia de muestreo = 100

Observe la señal de filtro y determine los coeficientes del filtro ButterWorth.



APÉNDICE B

CODIGOS DE LOS PROGRAMAS UTILIZADOS EN EL AMBIENTE GUI.

ARCHIVO jet_pds

```
acercate=figure(2);
set(acercate,'MenuBar','none','NumberTitle','off',...
    'Name','AMBIENTE GUI INTERACTIVO PARA EL MANEJO DE PROCESAMIENTO DE
SEÑALES',...
    'color',[0,0,1]);
load logo
h=surfl(L,source);
colormap(M);
ax=[0 60 0 60 -0.5 1];
ax=[7 52 7 52 -.5 .8];
axis(ax);
axis off;
shading flat;

m=moviein(24);

for n=1:24,
    rotate(h,[0 90],15,[21 21 0]);
    h=surfl(get(h,'XData'),get(h,'YData'),get(h,'ZData'),source);
    axis(ax);
    axis off;
    shading flat;
    m(:,n)=mvframe(0,24);
end;
uicontrol('Position',[0.1,0.76,0.8,0.23],'Units','Normalized','Style','Text',
    'HorizontalAlignment','Center','String',...
    {'AMBIENTE GUI INTERACTIVO PARA EL MANEJO DE PDS',...
    'por: JOSE EDWIN TORRES',...
    'Director: ING. CARLOS SANCHEZ',...
    '',...
    'FACULTAD DE INGENIERIA ELECTRICA',...
    'UNIVERSIDAD DE LA SALLE',...
    'BOGOTA, 2004'},...
    'BackgroundColor',[1,1,0],'ForegroundColor',[0,0,1]);
uicontrol('Position',[0.49,0.05,0.1,0.05],'Units','Normalized','Style','PushButton',
    'String','Continuar','Callback','if(exist('ventana')==0),Closereq,jet_princ,else,Closereq,end',...
    'BackgroundColor',[0,0,1],'ForegroundColor',[1,1,0]);
```



ARCHIVO jet_configuración

```
cla(vista1);
cla(vista2_1);
cla(vista2_1);
switch(get(practica, 'Value'))
case 1
    set(vista1, 'visible', 'on');
    set(vista2_1, 'visible', 'off');
    set(vista2_2, 'visible', 'off');
    set(text_1, 'String', 'Vpp');
    set(in_1, 'Visible', 'on', 'String', '180');
    set(text_2, 'String', 'Frecuencia');
    set(in_2, 'Visible', 'on', 'String', '5');
    set(text_3, 'String', 'Vdc');
    set(in_3, 'Visible', 'on', 'String', '15');
    set(text_4, 'String', 'Simetria');
    set(in_4, 'Visible', 'on', 'Value', 1, 'String', {'Impar', 'Par'});
    set(text_5, 'String', '');
    set(in_5, 'Visible', 'off');
    set(text_6, 'String', '');
    set(in_6, 'Visible', 'off');
case 2
    set(vista1, 'visible', 'on');
    set(vista2_1, 'visible', 'off');
    set(vista2_2, 'visible', 'off');
    set(text_1, 'String', 'Inicio de h(n)');
    set(in_1, 'Visible', 'on', 'String', '1');
    set(text_2, 'String', 'Inicio de x(n)');
    set(in_2, 'Visible', 'on', 'String', '0');
    set(text_3, 'String', '');
    set(in_3, 'Visible', 'off');
    set(text_4, 'String', '');
    set(in_4, 'Visible', 'off');
    set(text_5, 'String', 'Señal de entrada x(n)');
    set(in_5, 'Visible', 'on', 'String', '[1,2,3,1]');
    set(text_6, 'String', 'Comportamiento del sistema h(n)');
    set(in_6, 'Visible', 'on', 'String', '[1,2,1,-1]');
case 3
    set(vista1, 'visible', 'off');
    set(vista2_1, 'visible', 'on');
    set(vista2_2, 'visible', 'on');
    set(text_1, 'String', '');
    set(in_1, 'Visible', 'off', 'String', '200');
    set(text_2, 'String', '');
    set(in_2, 'Visible', 'off');
    set(text_3, 'String', '');
    set(in_3, 'Visible', 'off');
    set(text_4, 'String', '');
    set(in_4, 'Visible', 'off');
    set(text_5, 'String', 'Señal de entrada');
    set(in_5, 'Visible', 'on', 'String', '[2,2,2,2,2,2,2,2,2,2,2,2]');
    set(text_6, 'String', '');
```




```
set(in_6, 'Visible', 'off');
case 4
set(vista1, 'visible', 'off');
set(vista2_1, 'visible', 'on');
set(vista2_2, 'visible', 'on');
set(text_1, 'String', 'Muestras en frecuencia');
set(in_1, 'Visible', 'on', 'String', '207');
set(text_2, 'String', '');
set(in_2, 'Visible', 'off');
set(text_3, 'String', '');
set(in_3, 'Visible', 'off');
set(text_4, 'String', '');
set(in_4, 'Visible', 'off');
set(text_5, 'String', 'Señal de entrada');
set(in_5, 'Visible', 'on', 'String', '[2,2,2,2,2,2,2,2,2,2,2,2]');
set(text_6, 'String', '');
set(in_6, 'Visible', 'off');
case 5
set(vista1, 'visible', 'on');
set(vista2_1, 'visible', 'off');
set(vista2_2, 'visible', 'off');
set(text_1, 'String', '');
set(in_1, 'Visible', 'off');
set(text_2, 'String', '');
set(in_2, 'Visible', 'off');
set(text_3, 'String', '');
set(in_3, 'Visible', 'off');
set(text_4, 'String', '');
set(in_4, 'Visible', 'off');
set(text_5, 'String', 'Numerador');
set(in_5, 'Visible', 'on', 'String', '[3,1]');
set(text_6, 'String', 'Denominador');
set(in_6, 'Visible', 'on', 'String', '[3,4,56,21,87,1]');
case 6
set(vista1, 'visible', 'on');
set(vista2_1, 'visible', 'off');
set(vista2_2, 'visible', 'off');
set(text_1, 'String', '');
set(in_1, 'Visible', 'off', 'String', '');
set(text_2, 'String', '');
set(in_2, 'Visible', 'off');
set(text_3, 'String', '');
set(in_3, 'Visible', 'off');
set(text_4, 'String', '');
set(in_4, 'Visible', 'off');
set(text_5, 'String', 'Numerador');
set(in_5, 'Visible', 'on', 'String', '[1]');
set(text_6, 'String', 'Denominador');
set(in_6, 'Visible', 'on', 'String', '[1,-1,1]');
case 7
set(vista1, 'visible', 'on');
set(vista2_1, 'visible', 'off');
set(vista2_2, 'visible', 'off');
set(text_1, 'String', 'Orden del filtro');
```



```
set(in_1,'Visible','on','String','7');  
set(text_2,'String','Frecuencia de corte');  
set(in_2,'Visible','on','String','500');  
set(text_3,'String','Frecuencia de muestreo');  
set(in_3,'Visible','on','String','4000');  
set(text_4,'String','');  
set(in_4,'Visible','off');  
set(text_5,'String','');  
set(in_5,'Visible','off');  
set(text_6,'String','');  
set(in_6,'Visible','off');  
end
```

ARCHIVO jet_texto

```
switch(get(practica,'Value'))  
case 1  
    abajo={'                                     COMPONENTE DE  
    SEÑAL',...  
        ' ',...  
        'Vpp                                     introduzca la amplitud de la señal de  
pico a pico',...  
        'Frecuencia                             introduzca la frecuencia (Hz)',...  
        'Vdc                                     introduzca el nivel DC (constante) de la  
señal',...  
        'Simetria                               escoja la simetria de la señal segun el  
eje vertical',...  
        'En la grafica:                         Vdc -> Linea verde',...  
        '                                       Vrms -> Linea roja',...  
        ''};  
case 2  
    abajo={'  
CONVOLUCION',...  
        ' ',...  
        'Inicio de h(n)                         introduzca el tiempo para el cual  
empieza la secuencia h(n)',...  
        'Inicio de x(n)                         introduzca el tiempo para el cual  
empieza la secuencia x(n)',...  
        'x(n)                                     introduzca la secuencia x(n) entre [ ]  
separados por coma',...  
        'h(n)                                     introduzca la secuencia h(n) entre [ ]  
separados por coma',...  
        ' ',...  
        '  
Ej.                                     x(n)=[1,2,3,1]      h(n)=[1 ,2,1,-1]',...  
        '                                     para  
los respectivos      n=      0 1 2 3      -1 0 1 2'}];  
case 3  
    abajo={'                                     TRANSFORMADA  
DE FOURIER',...  
        ' ',...  
        'Señal de entrada                       Ingrese la secuencia de la señal de  
entrada entre [ ] separando cada dato por coma (,)',...  
        ''};
```



```
'',...
'Ej.          [2,2,2,2,2,2,-2,-2,-2,-2,-2,-2]',...
''};

case 4
    abajo={'                                     TRANSFORMADA
DISCRETA DE FOURIER',...
    '',...
    'Muestras en frecuencia          Cuantas muestras seran
tomadas del espectro',...
    'Señal de entrada          Ingrese la secuencia de la señal de
entrada entre [ ] separando cada dato por coma (,)',...
    '',...
    'Ej.          Muestras en frecuencia          =          200',...
    'Ej.          [2,2,2,2,2,2,-2,-2,-2,-2,-2,-2]',...
    ''};

case 5
    abajo={'                                     TRANSFORMADA DE
LAPLACE',...
    '(DIAGRAMA DE BODE)',...
    'ingrese la funcion de transferencia en funcion de S',...
    '',...
    'Ej.          S^2+3S+7          se escribe
[1,3,7]',...
    '',...
    'ingrese en cada la caja correspondiente, el numerador y el
denominador',...
    ''};

case 6
    abajo={'                                     TRANSFORMADA
Z',...
    '(DIAGRAMA DE NYQUIST)',...
    'ingrese la funcion de transferencia en funcion de Z',...
    '',...
    'Ej.          1+3Z^-1+7Z^-2          se escribe
[1,3,7]',...
    '',...
    'ingrese en cada la caja correspondiente, el numerador y el
denominador',...
    ''};

case 7
    abajo={'                                     FILTROS
DIGITALES',...
    '',...
    'Orden          introduzca el orden seguido por el
filtro',...
    'Frecuencia corte          introduzca la frecuencia de corte',...
    'Frecuencia de muestreo          introduzca la frecuencia de
corte',...
    '',...
    '          recuerde el teorema de Nyquist de muestreo de
señales'};
    set(text_4,'string','Coeficientes :');
    set(in_4,'visible','on','string',{num2str(P)});
end
```



```
set(texto, 'string', abajo);
```

ARCHIVO jet_calculos

```
%tesis
switch(get(practica, 'Value'))
case 1
    cla(vistal);
    Vpp=str2num(get(in_1, 'String'));
    f=str2num(get(in_2, 'String'));
    Vdc=str2num(get(in_3, 'String'));
    if(get(in_4, 'Value')==1)
        fase=0;
    else
        fase=pi/2;
    end
    t=-0.5:0.005:0.5;
    axes(vistal);
    plot(t, Vpp*sin(2*pi*f*t+fase)+Vdc)
    hold on
    patch([0,0], [-Vpp+Vdc, Vpp+Vdc], 'b-');
    plot(t, sqrt(2)*Vpp/2, 'r-')
    plot(t, Vdc, 'g-')
    hold off
case 2
    nh=str2num(get(in_1, 'String'));
    nx=str2num(get(in_2, 'String'));
    x=str2num(get(in_5, 'String'));
    h=str2num(get(in_6, 'String'));
    z=[0, conv(x,h), 0];
    n=0:length(z)-1;
    n=n+nh-nx-1;
    axes(vistal);
    stem(n,z);
case 3
    if(exist('n')==1)
        clear n;
    end
    if(exist('X')==1)
        clear X;
    end
    if(exist('x')==1)
        clear x;
    end
    x=str2num(get(in_5, 'String'));
    N=1000;
    n=0:length(x)-1;
    for(w=-pi:2*pi/N:pi-2*pi/N)
        suma=cumsum(x.*exp(-j*w*n));
        X(fix((w+pi)*N/2/pi)+1)=real(suma(1,length(suma)));
    end
    w=linspace(-pi,pi,N);
    axes(vista2_2);
    plot(w,X);
```



```
axes(vista2_1);
stem(n,x,'r-')
case 4
    if(exist('n')==1)
        clear n;
    end
    if(exist('X')==1)
        clear X;
    end
    N=str2num(get(in_1,'String'));
    x=str2num(get(in_5,'String'));
    n=0:length(x)-1;
    for(w=-pi:2*pi/N:pi-2*pi/N)
        suma=cumsum(x.*exp(-j*w*n));
        X(fix((w+pi)*N/2/pi)+1)=real(suma(1,length(suma)));
    end
    w=linspace(-pi,pi,N);
    axes(vista2_2);
    stem(w,X);
    axes(vista2_1);
    stem(n,x,'r-')
case 5
    num=str2num(get(in_5,'String'));
    den=str2num(get(in_6,'String'));
    w=logspace(0,5,60);
    [mag,ang]=bode(num,den,w);
    axes(vista1);
    semilogx(w,20*log10(mag))
case 6
    num=str2num(get(in_5,'String'));
    den=str2num(get(in_6,'String'));
    [re,im]=nyquist(num,den);
    axes(vista1);
    plot(re,im,'b-',re,-im,'b-')
case 7
    N=str2num(get(in_1,'String'));
    Wc=str2num(get(in_2,'String'));
    Wm=str2num(get(in_3,'String'));
    Wn=Wc/Wm;
    [B,A] = butter(N,Wn);
    [Z,P,K] = butter(N,Wn);
    [H,F]=freqz(B,A,512,Wm);
    axes(vista1);
    plot(F,abs(H));
    jet_textos
end
```



ARCHIVO jet_ayudas

```
switch(get(practica, 'Value'))
case 1
    help ops
    helpwin plot
    % ops
    % plot
    % sqrt
case 2
    helpwin conv
    % conv
case 3
    helpwin fourier
    % fourier
    % ifourier
case 4
    helpwin fft
    % dft
    % idft
    % fft
    % ifft
case 5
    help laplace
    helpwin bode
    % laplace
    % bode
case 6
    help ztrans
    helpwin nyquist
    % ztrans
    % iztrans
    % nyquist
case 7
    help freqz
    helpwin butter
    % butter
    % freqz
end
```



ARCHIVO jet_princ

```
clear all
clc
ventana=figure(199);
set(ventana,'MenuBar','none','NumberTitle','off',...
    'Name','AMBIENTE GUI INTERACTIVO PARA EL MANEJO DE PROCESAMIENTO DE
    SEÑALES',...
    'color',[0,0,1]);
escudo=axes('Position',[0.052,0.89,0.85,0.068],'Units','Normalized');
    image(imread('escudo','bmp'));
set(escudo,'XTickMode','manual','XTick',0,'YTickMode','manual','YTick',0,
    ...

    'XTickLabelMode','manual','XTickLabel','','YTickLabelMode','manual','YTic
    kLabel','');
    set(get(escudo,'xlabel'));
practica=uicontrol('Position',[0.25,0.71,0.65,0.15],'Units','Normalized',
    'Style','popupmenu','BackgroundColor',[1,1,1],'String',...
    {'COMPONENTES DE LAS SEÑALES','CONVOLUCION','TRANSFORMADA DE FOURIER
    CONTINUO','TRANSFORMADA DE FOURIER DISCRETA',...
    'TRANSFORMADA DE LAPLACE','TRANSFORMADA Z','FILTROS
    DIGITALES'},'Value',1,'Callback','jet_configuracion,jet_calculos,jet_text
    os');
vistal=axes('ycolor',[1 1 0],'xcolor',[1 1 0],'color',[1 1
0.7],'Position',[0.271,0.28,0.7,0.5],'Units','Normalized');
    get(vistal)
    plot(0,0)
    get(vistal)
    set(vistal,'color',[1,1,0.7]);
    get(vistal)
etiqx=uicontrol('Position',[0.271,0.78,0.7,0.03],'Units','Normalized','St
yle','Text','String','SEÑALES PARES, IMPARES, PERIODICAS,DC,AC,RMS',...

    'BackgroundColor',[0,0,1],'ForegroundColor',[1,1,0],'HorizontalAlignment
    ','center');
etiqx=uicontrol('Position',[0.271,0.22,0.7,0.03],'Units','Normalized','St
yle','Text','String','Tiempo',...

    'BackgroundColor',[0,0,1],'ForegroundColor',[1,1,0],'HorizontalAlignment
    ','center');

vista2_1=axes('Position',[0.271,0.549,0.7,0.227],'Units','Normalized','Vi
sible','off','color',[1,1,0.7]);
vista2_2=axes('Position',[0.271,0.28,0.7,0.227],'Units','Normalized','Vis
ible','off','color',[1,1,0.7]);
texto=uicontrol('Position',[0.271,0.046,0.699,0.169],'Units','Normalized'
,'Style','text','BackgroundColor',[1,1,0.7],...
    'HorizontalAlignment','Left');

frame=uicontrol('Position',[0.023,0.03,0.183,0.828],'Units','Normalized',
'Style','frame',...
    'BackgroundColor',[1,1,0]);
```



```
text_1=uicontrol('Position',[0.0273,0.8051,0.1465,0.0295],'Units','Normalized','Style','Text','HorizontalAlignment','Left','String','Vpp',...  
    'BackgroundColor',[1,1,0],'ForegroundColor',[0,0,1]);  
in_1=uicontrol('Position',[0.0586,0.75,0.1,0.03],'Units','Normalized','Style','Edit','String','180','CallBack','jet_calculos',...  
    'BackgroundColor',[1 1 1]);  
text_2=uicontrol('Position',[0.0273,0.7068,0.1465,0.0295],'Units','Normalized','Style','Text','HorizontalAlignment','Left','String','Frecuencia',...  
    ...  
    'BackgroundColor',[1,1,0],'ForegroundColor',[0,0,1]);  
in_2=uicontrol('Position',[0.0586,0.6531,0.1,0.03],'Units','Normalized','Style','Edit','String','5','CallBack','jet_calculos',...  
    'BackgroundColor',[1 1 1]);  
text_3=uicontrol('Position',[0.0273,0.6004,0.1465,0.0295],'Units','Normalized','Style','Text','HorizontalAlignment','Left','String','Vdc',...  
    'BackgroundColor',[1,1,0],'ForegroundColor',[0,0,1]);  
in_3=uicontrol('Position',[0.0586,0.5481,0.1,0.03],'Units','Normalized','Style','Edit','String','15','CallBack','jet_calculos',...  
    'BackgroundColor',[1 1 1]);  
text_4=uicontrol('Position',[0.0273,0.4765,0.1465,0.0295],'Units','Normalized','Style','Text','HorizontalAlignment','Left','String','Simetria',...  
    'BackgroundColor',[1,1,0],'ForegroundColor',[0,0,1]);  
in_4=uicontrol('Position',[0.0586,0.4284,0.1,0.03],'Units','Normalized','Style','popupmenu','String',{'Impar','Par'},'CallBack','if(get(practica,'value')==1),jet_calculos,end',...  
    'BackgroundColor',[1 1 1]);  
text_5=uicontrol('Position',[0.0273,0.3739,0.1465,0.0295],'Units','Normalized','Style','Text','HorizontalAlignment','Left','String','',...  
    'BackgroundColor',[1,1,0],'ForegroundColor',[0,0,1]);  
in_5=uicontrol('Position',[0.0586,0.341,0.1,0.03],'Units','Normalized','Style','Edit','Visible','off','String','', 'CallBack','jet_calculos',...  
    'BackgroundColor',[1 1 1]);  
text_6=uicontrol('Position',[0.0273,0.2770,0.1465,0.0295],'Units','Normalized','Style','Text','HorizontalAlignment','Left','String','',...  
    'BackgroundColor',[1,1,0],'ForegroundColor',[0,0,1]);  
in_6=uicontrol('Position',[0.0586,0.2430,0.1,0.03],'Units','Normalized','Style','Edit','Visible','off','String','', 'CallBack','jet_calculos',...  
    'BackgroundColor',[1 1 1]);  
  
ayuda=uicontrol('Position',[0.06,0.1685,0.1,0.05],'Units','Normalized','Style','PushButton','String','Ayuda','CallBack','jet_ayudas',...  
    'BackgroundColor',[0,0,1],'ForegroundColor',[1,1,0]);  
acerca=uicontrol('Position',[0.06,0.11,0.1,0.05],'Units','Normalized','Style','PushButton','String','Acerca','CallBack','jet_pds',...  
    'BackgroundColor',[0,0,1],'ForegroundColor',[1,1,0]);  
cerrar=uicontrol('Position',[0.06,0.05,0.1,0.05],'Units','Normalized','Style','PushButton','String','Cerrar','CallBack','Closereq,clear',...  
    'BackgroundColor',[0,0,1],'ForegroundColor',[1,1,0]);  
jet_calculos
```




APÉNDICE C

Instrucciones de instalación para el AMBIENTE GUI Interactivo en MATLAB.

1. Coloque el disco del AMBIENTE GUI (**Interactivo**) en la unidad lectora de 1.44 MB de su computadora.
2. Haga doble clic sobre el icono de *MI PC* que se encuentra en el *escritorio*, o un clic con el botón derecho del *ratón* y seleccione **abrir**.
3. Una vez abierta la carpeta de *MI PC* localice la unidad donde está insertado el disco **.Interactivo**. y ábralo con doble clic.
4. Simultáneamente, abra otra ventana desde *MI PC*. Unidad de disco duro (**C:**) y localice la carpeta donde se encuentran instalados los archivos de MATLAB, regularmente se encuentra en C:/MATLAB/... (el nombre puede variar ligeramente de acuerdo a la versión que se tiene instalada).
5. Dentro de la carpeta de MATLAB abra la carpeta **TOOLBOXES**.
6. Copie íntegramente la carpeta **AMBIENTE GUI**, ubicada en el disco de instalación, a la carpeta **TOOLBOXES**.
7. Ejecute el programa MATLAB y dentro del menú **.File.**, seleccione **.Set Path....**, se abrirá una nueva ventana llamada **.Path Browser**.
8. En el menú **.Path.**, seleccione **.Add to Path**.
9. Nuevamente se abrirá otra ventana, donde se debe escribir la trayectoria de la carpeta que desea añadir, en este caso **AMBIENTE GUI**, si no sabe de memoria la ubicación de la carpeta, de un clic sobre el botón que tiene tres puntos [...], podrá navegar por los directorios hasta encontrar y seleccionar la carpeta **AMBIENTE GUI**.
10. Para asegurar que la próxima vez que abra MATLAB no se haya borrado la trayectoria establecida, en el menú **.File**. Del **.Path Browser**. seleccione **.Save Path**. y cierre la ventana.
11. Para acceder al AMBIENTE GUI Interactivo, en la ventana de comando de MATLAB teclee: **jet_pds**



12. Con ése comando se abre la ventana de *presentación*, se pulsa el boton continuar e inmediatamente ingresa al ambiente GUI.



ANEXO A

Disco donde se encuentra la carpeta de **AMBIENTE GUI INTERACTIVO.**